The Romantic Voxspell Story

A spelling-aid program for second language learners

Kenney Chan
University of Auckland
Software Engineering, Department of Electrical and Computer Engineering
Auckland, New Zealand
Kcha582@aucklanduni.ac.nz

***Abstract*—The Romantic Voxspell Story is a spelling aid application integrated with story-telling aimed at second language learners, with the intention to help improve their spelling. (*Abstract*)**

Keywords— Voxspell, festival, spelling-aid, storytelling, challenge, spelling, software, second-languge learners. (key words)

I. Introduction

The Romantic Voxspell Story is a spelling aid application aimed at second language learners of age range 18 - 25 years old. The application integrates storytelling and gamification with spelling to maintain the user's attention, providing an educational yet entertaining way to practice spelling. The story is written with comedic elements with a cheerful atmosphere, as constantly spelling words can be tedious, and so, the integration of storytelling provides an entertaining way to learn to spell. Romance is the theme of the application, and the design of the UI is designed to match the story of the application, it is designed with the intention of simplicity, so that the functionality is intuitive. The application is user friendly and provides 2 different ways to practice spelling, users are given options to aid their understanding of using the spelling aid application, and a feedback system is implemented to allow the user to keep track of their progress. The intention of this report is to highlight and evaluate the process of develop of the application.

II. Graphical User Inteface (GUI)

A. Choice of programming language and packages used

The choice of programming language was Java at version 8. Java is a widely used programming language and has a difficulty of mediocrity with bountiful capabilities. It was chosen so that developers can understand the structure and implementation of the application with ease. It so allowed rapid implementation of the application due to its occasional simplicity. Java was also the most convenient language available at hand due to fresh and in depth exposure to the language.

The main package used was the VLCJ [1] package for the video playing ability of the application. The simplicity of installation and implementation of the package encourage the usage of it. The functionality is minimalistic yet provided a smooth and beautiful way to play videos.

Figure 1. Menu panel

B. Colour consideration

The colour consideration, and display layout was designed to link to the theme of romance and comedy in the story. Warm colours were used for the welcome panel, menu, and main statistics panel, this is to maintain attention and interest from the user as dull colours can be monotonous. The main colour used for these panels were pink as shown in Figure 1. Different colours were used for different elements of the quiz panel. For example, The progress bar was to represent love intuitively and so red was used for the like meter as the colour to convey its intentions. A duller colour was more appropriate for the attempt meter as this progress bar represents the chances the user still has, insinuating negativity. Different symbols with different colours were also used to describe the intention of each element in the quiz panel. Dark navy was used as the background colour of the option panel and statistics panel accessed from the quiz, the texts were also coloured white. Another theme the application is based on when it is not romantic is contemporary. An example of contemporary can be seen in the option, help and statistics panel accessed from the story quiz panel. The colour of these panels are dark navy. Meanwhile it contrasts the abundant amount of pink, it relieves the user from visual stress. It also prevents the application from being overwhelmed with pink by adding a fresh twist. Shades of lighter pink are used instead of brighter shades to prevent stress on the user's eyes. Colour

blindness was also considered in the selection of colours, the application scarcely uses green as the application is dominant in pink and red. Instead of green, blue or navy are used instead.

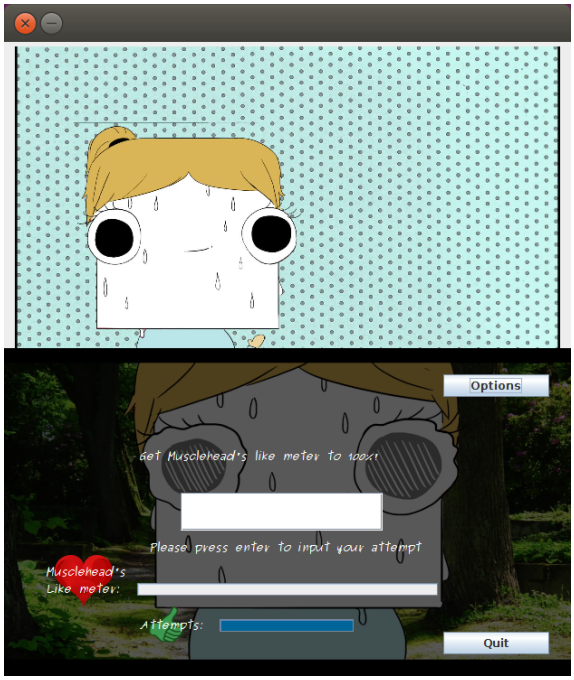

Figure 2. Quiz panel

C. Display Layout and Presentation of information

Display layout of each panel were spaced out to prevent confusion between elements, the number of elements were also kept minimal to keep the application intuitive. To replace the simplicity, each element is designed to be as aesthetically pleasing as possible. The GUI changes panels whenever the user presses a button, the panel is resized it requires, only the "options panel", "the select a list" button, have prompts and warning in separate windows to the applications. This is to prevent the application overloading the user's computer with multiple frames.

Referring to figure 2, the layout of the quiz panels for both story and free play mode draws inspiration from the Nintendo DS and dating simulators, as the video panel on the top half play videos of the character speaking, while you interact with them on the bottom half of the screen, this creates an entertaining and innovative experience to practicing spelling. Although it may be stressful and time consuming to constantly look up and down between the video panel and the text box, the program is timed to prevent stress on the user.

Instead of using buttons provided by Swing, they are replaced by icons of photoshopped hearts or drawn images. Images of hearts are cumulated from royalty free image websites [3]. For example, referring to figure 3, different episodes are represented by an icon of the character you are helping. Using custom made buttons are more aesthetically impressive than buttons provided by Swing, this makes the using the application satisfying, as boring aesthetics may bore the user. Though buttons that need to be intuitive and simple uses the buttons that swing provides, these buttons are usually unimportant, and does not contribute to the overall aesthetics. All the videos made in the application are drawn and edited by the developer Kenney Chan.

The application exercises the use of Swing components to display instinctive presentation of each informative element. For example, as seen in figure 2, progress bars are used to keep track of progress of the current game, tables are used to display information that needs differentiation, and selection between elements are conveyed through combo boxes. The functionality of these components are favoured over aesthetics, therefore Swing components are used instead of custom made components. All of these functionality of these components are kept as simple as possible, they are sized and positioned where they could be easily found.

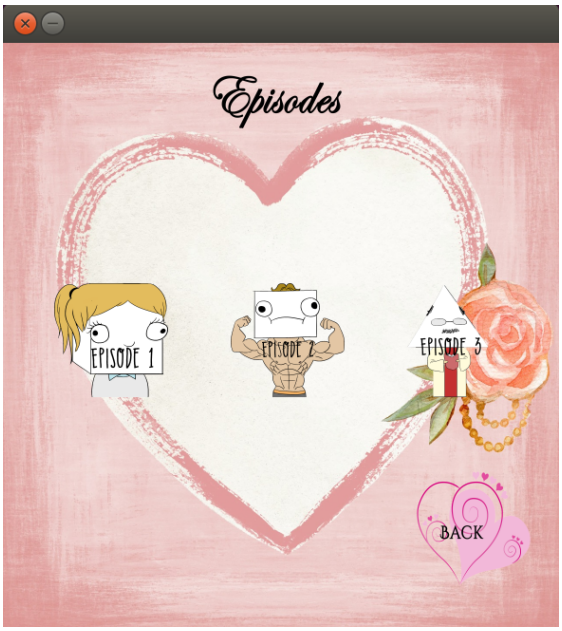

Figure 3. Episode selection panel

D. Other presentation issues

The story is conveyed through the use of videos, even for short videos like when the character is speaking. This is chosen over the use of GIF images to provide a higher resolution and fluidity, though, this consequently made the application unnecessarily large in size. During the development of the application, users are considered to have enough memory to store the application, so the fluidity, and aesthetics are favoured over the size.

The size of the frame is not resizable as this will ruin the resolution of the images and video, another alternative is to add white space into the frame, but this will most definite ruin the aesthetics of the application. The frame is coded to a reasonable size which allows the user to read words and watch the images clearly.

III. FUNTIONALITY

The functionality of the application focuses heavily on the intention of improving the spelling ability of the user. Constantly spelling and staying on the same interface can be tedious, and implementing challenges and achievements is cliché, therefore, integrating storytelling into the application was the solution to create an educational yet entertaining application. As the target audience are second language learners in the range of 18 to 25 years old, it is difficult for them to focus assuming people in this age have constant and rapidly changing lives.

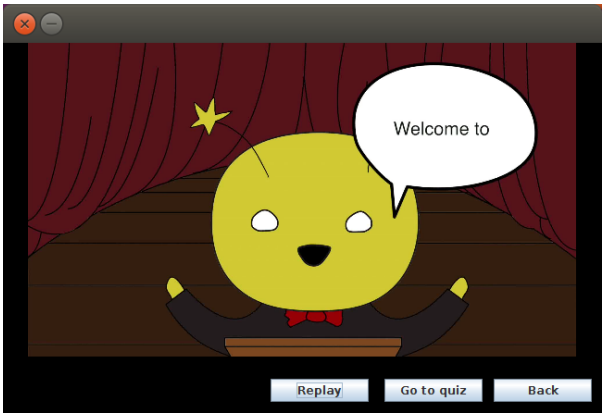

Figure 4. Movie panel

A. Story mode.

During the planning of the product, there were many established ways to integrate the story telling functionality. These include filling in the blanks of incomplete text stories, or spelling words while a voice reads a story. The final decision was concluded on video based storytelling, as filling in visual components and listening to audio can be monotonous, combining these two into animated video with sound will provide a much more entertaining substitution.

The story is separated into 3 episodes, the intention of this is to divide the difficulty of the words into three different levels. This also allows the user to progress in small intervals. This maintains the attention of the user as each episode fills the user with suspense. Each episode starts off with a movie played in a movie panel as shown in figure 4. Making a good impression is important, therefore an amusing animation that starts the story off before an entertaining spelling quiz will most definitely capture the user's attention. Due to the target audience and a comedic theme, animations are created instead of live videos as this gives much more creative freedom, and the development on comedy value is easier. After each quiz, the introduction video for the next episode automatically plays. If the user does not want to move to the next episode, they can press the back button on the movie panel as seen in figure 4.

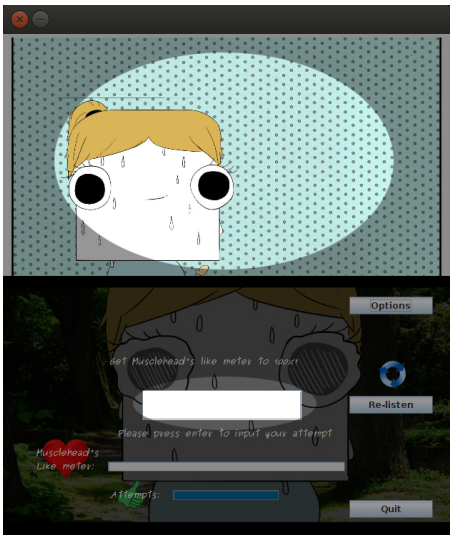

Figure 5. User and application interaction

The application puts a twist on the functionality by interacting with the user. The user is given the task to help spell words that a character in the story cannot. They are required to fill in a "like meter" in order to progress with the story as the character they are helping is trying to impress another character. The "like meter" also displays sentences that convey the thoughts of the character you are trying to help impress. This is to add comedic value to the story.

The application uses a text to speech application called Festival [5] to tell the user then word to spell. It also tells the user if the user has spelled the word correctly or incorrectly. Festival [5] is a simple and versatile, open source speech synthesis application, therefore it is suitable for the use of spelling aid. Also the monotonous default voice adds comedic value to the application. The application plays a background music obtained from royalty free music websites [4] during the quiz to accompany the game, this makes the game more entertaining as a game without music can feel incomplete. An option is available for the user to turn off the music if they find it distracting.

The application interacts by chaining the words the user is spelling with the words the character is saying. In figure 5, the upper screen displays a picture of the character the user is helping, it will play videos of the character speaking. The user interacts with the character by spelling and submitting words in the text box on the bottom half of the screen. If the character says "hello" and the user is asked to spell "there", the character intends to say "hello there", but needs the help of the user. The purpose of this interaction is provide the user a purpose to spell and to make progress. The application encourages the user to spell words correctly, ultimately sustaining the user's attention while educating them.

The user has an option to re-listen to the word again, but only once, the re-listen button will disappear after pressing it. The user is also given in total three chances to spell words incorrectly, reason for these harsh condition is to make the story more realistic as love and relationships do not come easily, this also proposes a challenge to the user, adding another element of

surprise to the application. The user is able to keep track of their chances in the “Attempts” bar.

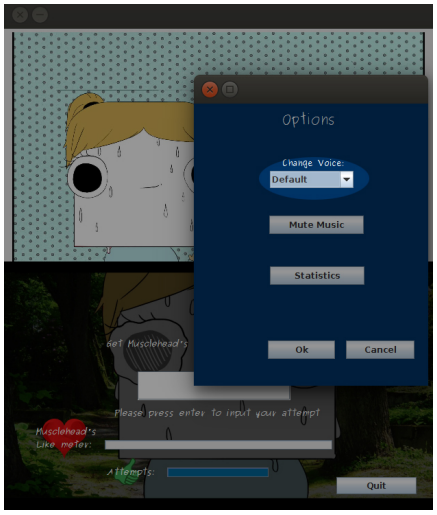

Figure 6. Options window

An options button is also available for the user to change the settings of the application. This includes changing the voice, because the target audience is aimed at second language speakers, it is plausible to assume many users may have trouble understanding certain accents. The user must press ok to finalize their decision, or else the voice will not change, this is a fail-safe option in case the user has selected the incorrect voice. The user may also mute the music if it is too distracting or loud and they are unable to hear the word they are tasked to spell. The user is also given the option to look at the statistics of the current progress at runtime if they are curious.

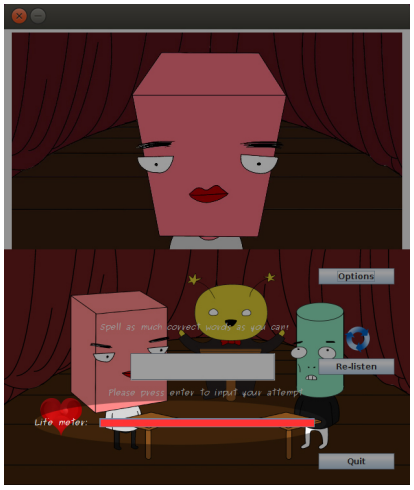

Figure 7. Free play quiz panel

B. *Free play mode*

When the user has finished the story and wishes to practice their spelling more or is not interested in the story, “Free play mode” is available. The story behind free play mode is that the user is in a TV show for a spelling bee, and the character must spell as much words as possible. Though the story will not progress, and only finishes when the user has spelt words incorrectly. Before the start of free play mode, the free play mode menu will appear to allow the user to insert a list of words they wish to quiz themselves on. A list of words that is to national New Zealand educational standard is provided for guidance and use. The user must also insert a list before they can start the game, they can choose their own list or the list provided. The reason for this is because there are no hard coded words in free play mode and there is no memory of word list selected from before, so the user must select a list before gameplay.

The gameplay for free play is the same as the story mode but has a different way of progression, it also plays an introduction video. Free play mode does not have any progressive storyline; the game only ends when the user gets 10 words incorrect. As seen in figure 7, the “like meter” is replaced with a “life meter” which indicates how many incorrect words the user may currently make. The reason for this functionality is to prevent the game from being too repetitive, as constantly spelling words may become tedious quickly, it also gives the user a reason to stop and retry. The top panel will be just an animation of a character talking, this character is presumably the user. A hint will also appear below the “life meter” if the user spells more than 3 incorrect words, the hint will reveal letters of the word based on the number of letters, and the amount of time the user gets the word incorrect. As there is no story to free play mode, the hint provides a medium for the user to continue spelling and aid them in learning to spell the words correctly. A random number (limited) of letters of the words are randomly replaced with an underscore “_”. While revealing some letters in their correct position. It is up to the user to fill in the missing letters.

An options panel is also available for the user to change the voice of festival [5], or mute background music. A different background will also be playing to match the theme of a spelling bee.

Figure 8. Statistics panel

C. *Statistics panel*

When the user wants to look at their record and history the can access the statistics panel from main menu.

Here the user will be able to access statistics for story mode and free play mode, they can change this by selecting their choice on the left combo box as seen in figure 8, this allows freedom and differentiates between custom words from manually inserted words. If the user chooses to look at statistics for story mode, another combo box will appear on the left, the user will be able to select which episode's statistics they would like to look at. These differentiation allows the user to keep track of their progress without confusion, the more sorted the information is, the easier it is to understand.

There are 3 information displayed on the table, the word, the number of times correct and the number of times incorrect. The word is crucial as the user must know the actual spelling, and which word to improve on. Using incorrect and correct are very intuitive. The user can improve on words that have a dominant number of incorrect tries. The use of mastered, faulted and failed was considered but it was concluded that this is confusing since it requires the user to understand the meaning of each record. The application runs on simplicity to keep the application instinctive to the user.

The storage of record is saved in a simple format inside text files, two separate files are required to for the story mode, and free play mode. This gives display of information easily

IV. Development and design of code

The coding of application is done on Eclipse, an integrated development environment (IDE). Eclipse was chosen as it provides a convenient environment to develop the program, it provides many functionalities to aid the development. Java was the chosen programming language.

A. The choice of Java

Java was used to code the application, and although Java is a powerful language, it is not the best choice. Gaming friendly programming languages such as C++ or C# would be suitable due to it being widely used, it's professionalism and coding convenience. During the development of the code, good practices were considered, but this was proven to be difficult as Java is a very versatile language. Coupled with the need of many functionality, good coding was hard to achieve. It is safe to say that the coding was implemented to the best quality it could achieve. The main coding practices focused include avoidance of static fields and the use of fields. The code could be improved by using encapsulation more as there are too many classes.

Inconveniently the user must also understand how to run jar files, follow unintuitive instructions, or learn how to run bash scripts. The user may also not have Java [6] installed on their system which could prevent the user from using that application without having to install extra applications. But if the user has java [6] installed, a convenience is that the application does not require installation which could potentially save the user time.

B. Model,view and control design pattern

The main design pattern used for the development of the code was the model, view and control pattern. This design pattern was achieved by creating model classes for all the quiz modes available, and view and control classes for every available panel. This model class controls the functionality of the quiz, this includes word comparisons, file IO, and updating the panels. The main Voxspell class is also a model class as this class controls how the panel changes. Each panel are implemented as a view and control class; the user interacts with the panel; the panel sends the action of the user to the main Voxspell class which changes the panel according to the action. This pattern is easy to implement and change, it also allows runtime changes from the user to be easily implemented. The separated presentation allows changes to be done conveniently, avoiding duplicate code. It makes a clear division between the real world and the application, while the code presents a medium for the application to work in a self-contained environment and support multiple presentation simultaneously.

C. Singleton

The application heavily relies speech synthesis to inform the user the correct word to spell. The design of the code has a Speaker class which will run whenever text to speech is required. This speaker class is a singleton. Singletons are convenient as the underlying application will not be overwhelmed with multiple threads running the speech synthesis application relieving stress on the user's computer. Singletons also prevent the usage of static modifiers, a bad coding practice.

D. Festival

The application exercises the use of speech synthesis to inform the user of the word to spell. The application uses a software called festival [5] to convert text to audio speech. To do this, the application feeds a sentence, and the chosen voice to a bash script that opens festival [5], speaks, and closes application. Festival [5] provides the application the opportunity to not require the user to set anything for speech synthesis to be functional. Festival [5] is also by default installed onto Ubuntu and Linux system, this is convenient for users who have this operating system. Inconveniently, this also means that the user must have festival [5] on their systems, which could cause people with OSX or Windows to unable to run the application with the speech synthesis. Without speech synthesis, this will render the application trivial. The user must then install festival [5] in order to obtain full functionality of the script.

E. VLCJ

The application relies heavily on videos to allow the application to interact with the user and to progress with the story. The application uses an external library called VLCJ [1] that provides the application the functionality to play videos. VLCJ [1] is straight forward and provides countless functionality. An important functionality is the ability to play multiple videos in order as videos of the character speaking is required for interaction. VLCJ [1] is open sourced so extra functionality provided by other users can be easily accessed and installed.

F. Java Swing

The GUI of the application is implemented by Java Swing. Swing is a widget toolkit that provides components for GUI designs. Swing provides an uncomplicated way to implement GUIs. Window Builder [2] is a Swing designer, and allows developers to create user interfaces with ease by providing drag and drop capabilities. This application was used to develop the

UI of the application due to its convenience and ability to generate code automatically. Adding custom icons and images was not a complicated process, coupled with simplicity, Swing provides simple implementation yet beautiful interfaces for applications. To allow user interaction, each components were embedded with listeners, as this adds functionalities and feeds the application information based on the choices the user selects.

G. The development process

The development of the application is separated into three stages: prototype, beta, and final.

The application prototype was implemented by the collaboration of two developers. The collaboration was well planned and the work load for each developers was evenly distributed. Development processes were used to produce a robust and quality assured code:

- Pair programming: During the development of the prototype of the application, it was created by two developers. Pair program was exercised to produce robust code. Pair programming is where one developer, the driver, writes the code, and another developer, the navigator, reviews the lines of code to find potential mistakes. This process is an agile software development technique as both the process of implementation and testing are done together.
- eXtreme Programming (XP): To produce code of quality the XP development process was used. XP is another agile development technique, the code is firstly planned, implemented, and finally tested, this processes repeats until code that is to standard is produced. Each changes to the code was also documented onto journals to keep track of changes.
- Waterfall model: Early implementation of the codes follows a simple development model. Establishing requirements, designing, implementation, verification, then finally maintenance. Establishing requirements and designing are done on paper where screen diagrams are used to create samples of the panels. Implementation is the done separately where

Once the prototype was complete, the beta version was done individually, this gave the developers creative freedom to direct the theme of the application however they are pleased. The beta version was also distributed to Class peers for feedback. Drawing inspiration from dating simulators and The Nintendo DS, the direction The Romantic Voxspell Story was developed heavily based on storytelling with theme of comedy and romance. The intention of the beta version was to give evaluators a "taste" of spelling aid integrated with storytelling. For this reason, only the first episode was available to the user, and many functionalities were not available. The beta version provided evaluators the introduction video, the interactive game play, and the result of failing to progress with the story. The intention of this is to get feedback without giving away the surprise the application proposes. The application is heavily based storytelling, so only feedback on the story telling element was desired.

After obtaining generally positive feedback, the direction of storytelling was proven to be very entertaining, so more episodes were implemented. More functionality was added based on the feedback from the beta version. After the implementation the application was tested by letting people of the targeted age range to play the application. Many of the implementation of the final application are extracted and improved to match the application's intended functionalities so many implementations of the code also share contribution from both the developers in the prototype versions.

H. Version Control

Keeping track and freedom of access to older versions of the code is important as past implementation may contain bugs or useless implementation that requires isolation. It also acts as a medium for backups in case if the file may be potentially lost. Version control also allows multiple developers to work on the application simultaneously which is required for the prototype of the application. Github was used as for version control.

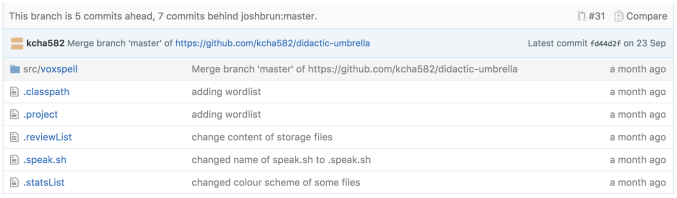

Figure 9. Github repository

During the development of the prototype, a repository was shared between the two developers. As shown in figure 9 the main repository was located at Joshua Brundan's computer, meanwhile the other developer Kenney Chan keeps a copy of the repository from forking. This decision was made as Kenney believes he is too clumsy to keep the main version of the repository.

Github was chosen for its simplicity and providing a safe and convenient solution to version control. Though because both the developers were not experienced with Github, time was taken to understand it's functionality, but it was ultimately proven to be very advantageous to use Github. Github is also widely used so potential future developers may access and use the repository with ease.

I. Journal

Other than version control planning and processes are recorded in a hand written journal. The hand written journal is a useful way to record thoughts that could be later expanded with other elements. It is also useful to record the date and time when an implementation is made or to record what is done on a specific day.

J. Shortcut keys

The application only contains two shortcut key. One is pressing enter when the user wants to submit a spelling attempt. This is a very widely used short cut key so excluding this

functionality will most definitely reduce the userability of the application. This shortcut key is also a very instinctive function as many other applications exercise the use of the enter (or return) button to submit information to the application. The implementation of this is done by adding an enter key listener to the text box. The use of a listener was the simplest way to add user interactions to the Swing component. The second shortcut key is the welcome panel. The user must press a key to move from the welcome panel to the menu panel. The reason for this is to prevent the user from accidentally pressing a button in the menu panel if a mouse is used to move panels instead. The implementation was done by adding a key listener to the welcome panel.

V. Evaluation and Testing

To produce a robust application of high quality, the application was repetitively tested to find potential bugs improvements. Evaluation was crucial in order to informed of the potential improvements that could be made to the application, evaluation was done after the completion of the beta and final versions.

A. Testing by oneself

To test the application for potential bugs the application was ran and played repetitively. All possible actions and inputs were tested to find bugs and verify that the application does what each functionality was intended to do. This method of finding bugs was effective as many bugs were found during this process.

B. Testing by class peers

Class peers were given a copy of the beta versions to evaluate and test the functionality of the code. 4 feedbacks were obtained from class peers. Fortunately, all four feedback state the sole same bug: a null pointer exception is printed on the console at certain times of the application. This problem was easily fixed as videos that was removed was accidentally implemented into the code. The lack of bugs from the feedback was a good sign that the code is robust. Though a reason for this could be because the beta version was too simple and lacked lots of functionality. Lacking chances for the class peers to find potential bugs.

C. Evaluation by oneself

To evaluate the application, the aesthetics and story line is constantly changed to match the theme of romance and comedy. An important process of thought is to imagine oneself as a user: "if I as a user, do not find the application to be attractive or comedic, then there is a big chance that others will also find it unattractive and dull". This evaluation was used to improve the quality of the application and was proven to be effective. Though it is hard to evaluate the application as feedback from oneself can be biased. Additionally, because every user favour thing differently it is important not to improve the application based solely on evaluation by oneself.

D. Evaluation by class peers

The evaluation from class peers received are generally positive, there were lots of compliments for creativity of the user-application interaction, and the use and choice of animations and audio. Though many stated that a lot of the functionality was missing, but this was intentional. A major feedback that was put into heavy consideration was the option to play the game without having a storyline, as some may only want a spelling aid application.

E. Solutions and changes to the problems

To fix the problems, print statements were mainly used to track where the bug lies, the flow of information was followed to find which class or method is causing the bug. Different input is then fed into the application to find how the bug affects the application and to find a solution to the bug.

The final application is different to the beta version by having much more functionality, like the rest of the story and the free play mode. The free play mode was ultimately implemented with spelling list selection freedom as the program is intended to be versatile and intuitive, and any chances of maintaining the user's attention were considered as a potential improvement. So, the major improvements made was the implementation of functionality.

F. Other evaluations

Other evaluations were made by having multiple target audience (that are not part of the software engineering cohort) play the application and provide feedback after using the application. The evaluation was done on both the beta and final version to provide unique feedback. This was proven to be effective as it is discovered that feedback from audiences that are not software engineers differs greatly. Meanwhile software engineers look for fluidity, convenience and robustness. It is discovered general audiences focus more on the evaluation of the aesthetics and creativity of the gameplay. Of course the application is targets all audience that lies within the correct age range, so judgement of evaluation was fair.

VI. Future work

There remain bountiful improvements the application can make:

- A gamification friendly programming language would be used such as C# instead of Java [5]. This is so that implementation of the application is easier.
- Functionality to add stories should be added. This is so that the application can have multiple themes, instead of heavily focusing on romance.
- A colour scheme of multiple colours instead of shades of pink can be used, this provides a much more attractive UI, and because of the heavy use of pink, many males may find the colour unsuitable.
- The feedback from the application would be improved inform the user of the correct spelling. The feedback of different form such as graphs would also be helpful.
- The use of a speech synthesis that is available to all operating system will be convenient to different kinds of users.
- A beta version of the application with more functionality should be distributed for evaluation

and testing. This is so that more useful information can be used.

- Improvement on the userability will most definite make the application more versatile to all different users.
- Different modes or episode that could allow the application to target multiple audience of different ages.
- More design patterns should be used to improve the quality of the code.

And others among the abundant amount of improvements…

VII. Conclusion

In conclusion The Romantic Voxspell Story is an application that integrates spelling aid with storytelling. A free play mode is available for the user if they choose to just practice spelling words. The application provides a feedback which allows the user to improve their spelling based on these feedbacks. The application is written in Java, and despite its capabilities, it was proven to be a complicated programming languages to make the application out of. Evaluation and testing were done for feedback on the application, the application receives generally positive criticism with useful potential improvements. Improvements to the application could be made such as improvement of the design of the code, the ability to insert multiple stories and a beta version with more functionality.

Acknowledgement

An acknowledgement to Joshua Brundan's contribution to the implementation of the prototype version of the application, which the final application is based on.

An acknowledgement to Tei Kim for the aid in producing the drawing that are used for the videos and buttons for the application.

References

[1] "VLCJ". *Caprica.github.io*. N.p., 2016. Web.
[2] "Window Builder". *Eclipse*. N.p., 2016. Web.
[3] "Free Images - Pixabay". *Pixabay.com*. N.p., 2016. Web.
[4] "Royalty Free Music". *Incompetech.com*. N.p., 2016. Web
[5] "Festival". *Cstr.ed.ac.uk*. N.p., 2016.
[6] "Java.Com: Java + You". *Java.com*. N.p., 2016.