IS51016A: Audio Visual Computing
Coursework 2 Report
Brock Craft (your student no.)

- Jacky Wang, Yuvesh Tulsiani

Executive Summary

We have created a Visual Novel for our Audio-Visual project as a way to showcase our familiarity with this form of digital media and to know what it takes to build a program that incorporates the features synonymous with what is usually found in other visual novels.

Features and User Interactions

A visual novel is a piece of digital media which uses pictures, music and text with some animation to deliver an interactive story experience which allows the user to play and explore the story at their own pace. We have decided to create a visual novel as to exercise our Audio-Visual programming skills, in order to at least develop a visual-novel engine that is capable of displaying characters, backgrounds, sounds and text in a coherent environment that is functionally similar to the likes of what is already available as a commercial product. Whilst visual novels aren't inherently focused on active user interaction, they are growing in popularity as a digital storytelling medium, with decisive and emotion-driven choices made by the user making the consequences of a story something the user feels responsible for.

Development

As we were working on the project as a pair we thought it would be best to use GitHub [1] as a tool to make sure we both had the most up to date code and it meant we have a changelog.

To create a visual novel, the usual format for an application like this would include a text box, which is the primary means of delivering the story with the pictures of the character and the background to supplement and enhance the storytelling experience, as well as suitable ambient soundtrack that would play alongside the story to enhance the mood of the scene. These audio and visual cues would become active in response to the text.

When we made ours, we used the text file that would read each line separately, which would become the entire content of each textbox [5]. Our application would then parse the line for any specific cues that could trigger a function for some music or animation that would happen

onscreen. This makes it easy to swap images, apply effects, or play certain sounds to specific points in the visual novel.

Some examples of the cues we have employed within this triggering system we have in place include screen-based shake effects, if the scene involves some form of perhaps a crash or a loud noise to name a few examples. Music could switch between different scenes, as well as ambient noises to further envelope the user within the story. Character entrances and exits are also one of the possible events, and we can incorporate different text styles in order for the player to differentiate from which character the text is coming from.

For our code, we used what we learnt to create a suitable base for our program, utilizing object-oriented design in a way which makes sense logically and should be fairly easy to modify and understand given our layout. We created our application alongside the features we felt were most important first, and went along the development process to add more and more features as we saw fit to try and replicate the same things we encountered upon our experiences with visual novels.

We have also used the Maxim audio library [2] in order for us to play sound in our project. As mentioned, we have complicated sounds that we need to utilize in our application, and thus, we require something that allows processing to cope with reading and using sound files.

To give an outline of how we put the application together, variables have been used to store values that might be modified by user input or functions later on, such as the state of the menu or the text progression index. Classes and functions have been used for sorting out large blocks of code so that we are able to call functions within that class without having to write it again or cluttering the setup function with unnecessary initializations only relevant to the class. We have used classes for the character, so that we are able to call a character's file and draw data from their respective files.

In the end, I believe we have achieved a good level of potential user experience in terms of providing a means of which to convey text-based stories with layers of visual and audio feedback, as well as allowing the player to insert themselves as a character in the story. We are, however, not professional storytellers, but feel as though what we have built is suitable as a base for a story not necessarily written by us, but able to tell a story comparatively with the same depth and feel of any other visual novel, as far as the core mechanics are in place.

References

Sites used when making the project

[1]
https://github.com/uveavanto/AVProject
GitHub was used for collaboration.

[2]
https://github.com/micknoise/Maxim
Maxim was used in our project so we could have audio playing in the background.

[3]
http://www.looperman.com/loops/detail/82039
Looperman was one of the sources used to get open source sound files to use as background music.

[4]
http://www.learningprocessing.com/examples/chapter-18/example-18-1/
This processing example where we learnt how to do basic text input.

[5]
https://processing.org/reference/loadStrings_.html
The processing reference was used countless times during the time we were working on our project. The loadStrings command is probably the most important page for our project.

[6]
http://processingjs.org/reference/link_/
We used the processing js reference as well as a lot of the things there carry over to the java version.