Project Whitekey

Project Evaluation

Andrew Pinion • **Jace Courville** • **Mitchell Mason**

ID:CS338037 ID:CS338008 ID:CS338031

What Went Right

- Setting a realistic goal: From the outset, we wanted a project that could be reasonably completed before the end of one semester. We decided to take one core feature and do it very well, as opposed to many features not so well. The result is a highly polished application that can be deployed on the Google Play Store in the immediate future.

- Our Concept: The idea is simple but its practical usage is powerful. The keyboard is easy enough to use by anyone, and it is also useful for improvisational performances by professionals.

- The User Interface: Although we have no budget, our user interface design far exceeds that of any competing app currently available in the Play Store.

- Concept validation: We performed a live demonstration of our application to Belle Chasse High School's band class and its instructor, and every user who tried it was able to make music with minimal to no effort. Several people who tried the application indicated that they would pay more than our asking price for it.

What Went Wrong

- Github: A substantial amount of time was consumed early in the development stage sorting out versioning problems and conflicts with git.

- Problems with Eclipse: A portion of Eclipse's auto-generated code for Android classes contains depreciated methods and errors, which was unexpected. Compiling the project was difficult initially due to these errors.

- Android layout problems: Eclipse's Android SDK plugin doesn't provide a true WYSIWYG editor for user interfaces, and drag support for controls is limited which made placement of the piano keys difficult and time consuming. Our initial usage of an Absolutelayout quickly became problematic.

- Android API sound quality issues: There is little documentation in terms of troubleshooting the audio API for Android, and popping sounds have been difficult to eliminate in our audio.

Things to Change

- Github/Version Control: Dealing with Github related issues was our largest setback and time sink, and if we redid this project from scratch better planning and setup for version control would be needed.

- Layout files: Creating the user interface with RelativeLayouts from the outset and editing raw XML as opposed to battling the GUI designer would have saved a significant amount of time.

Lessons Learned

- Eclipse has many issues: We encountered many bugs with the IDE throughout development, some of which appeared randomly and could not be reasonably explained.

- Marketing issues: Our application is not technically a piano, and educating users on the fact that this is much like a new instrument may prove difficult in the near future.

- Goal Setting: By choosing a goal that was realistic early in the project, we were able to create a highly polished application without a budget and within a limited timeframe.