Gina Filangeri
Josiah Pang
Celestine Co
Sharmayne Tanedo
CPE 123 MWF 10-12

60% of Final Project Update

Description:

For our final project in CPE 123 music class, we as a group decided to make a synthesizer-like program. On the bottom half of the space, we have all of the "sound makers." On the left is a regular piano keyboard that spans three octaves, and the right contains all the sound buttons such as drums, hi-hats and dings that can create cool remixes with what you are playing on the piano, or you can play over whatever you load as a backtrack.

You can use this program via touch screen, keyboard, or mousepad. We suggest using touchscreen or keyboard to simulate the experience most similar to an actual piano. On the top half, as of now you will see two square buttons: the load backtrack button and a stop button. With the load backtrack button, you are redirected to your finder files, where you can choose a wav file to play as a backtrack to the program. While the music is playing in the back, you can make your own personal remix of a song. We will also create a volume button that can control how loud the backtrack is, so that you can hear the piano keys and synthesizer notes over the backtrack. Since our program looks very simple and organized we were thinking that this could be an instrument oriented towards beginners with little to no musical knowledge and/or experience.

The final 40% of the project which we are currently working on is the looper, where we will create a button that will record whatever is being played, and then loop it in the background with the backtrack. Also, we are trying to work on a record button that records the backtrack with all the new sounds/remixes you created and saves it as a new wav file. We are also planning on using abstraction because we currently have a lot of repetition from the piano keys and sound board. Also, we will be working on having a beat (such as boots and cats) automatically be playing when the user opens the program so that it is not just silent. Of course the user may choose to turn off the automatic backtrack if they want to make their music all by themselves.

Prototype:

We initially created a picture through photoshop for aesthetic purposes for our synthesizer. We did this because it is much easier to load a single image than try to use "place-image" many times over and over again. With this picture, we assigned certain frequencies/functions for each button depending on their purpose. For the keyboard prototype, we created a conditional function (checkKey) that takes the dimensions of the image and divides it into the correct spaces for the keys. This checkKey function is linked to our playKey function, which simply plays the sound assigned (piano tone, drum, clap, etc.) to that given key. This being said, if a certain range of the x and y value in the board is pressed, our big bang function will play the sound of the key in that certain general range. The functions used in the keyboard prototype is similar to our sound booth button prototype- same functions used with different dimensions.

Our big bang function starts with our initial state, which is just the image of our synthesizer. The handle-mouse and handle-key functions are our interaction with our program, which allows us to press or touch the keys and buttons you want to play.

Testing

To test and receive feedback about our program we asked both engineer majors and non-engineer majors, although it did not really matter who we showed, because the program is largely music based and does not require technical knowledge. We consistently asked five questions to the testers: if they liked the way the program looked, if they liked the idea, if it was easy to use, if it was something they would use and play with in the future, and if they had any feedback or features that they would like to see added to the program.

All of our testers spent time playing around with the program for over five minutes. We explained to them what the program was and let them take it from there, and asked guiding questions along the way.

While they were playing with the program, we led them through a tutorial of how to use it, which buttons correspond to which key, and how to use the backtrack. When we tried to explain it to them, our program already was visually self-explanatory, so we just briefly explained what to do.

Analysis

The following is the compilation of all of the responses we got from the various people that tested our program.

- Aesthetic:
    - It was overall positively received. People agreed that it was pleasing to the eyes and most liked the color scheme. One person did mention it was confusing to an actual piano player, but after playing around with the program they got used to it. Many people would like to see the keys change color when pressed, and that will definitely be something we'll add to it.
- Do you like the idea:
    - All of our testers like the idea of a synthesizer-like program. They thought the idea of using a backtrack was cool. Many labeled the program as "fun" and "easy for even beginners."

- <u>Is it easy to use?:</u>
    - Everyone thought the keys were really easy to use. The fact that we color coded most of our board and labeled all of the keys and buttons to the keyboard symbols made it really easy to maneuver. Also, the fact that our program is similar enough to a keyboard, people thought it was easy to play small songs with it. Most also commented on the well organized aspect of the program and how labeling the keys really helps.
- <u>Is this something you would use?:</u>
    - Some people responded negatively to this. They would only use it for amusement, but if they actually want to create remixes of songs, one person said he "rather use GarageBand". Another person said he would use it if there were more keys rather than just three octaves. However someone said that they prefer our version of the keyboard to the one in GarageBand; they thought it was a simpler and efficient version of GarageBand. Despite that, we had someone say that since he was not at all music oriented, he wouldn't think to use something like this, but after trying it out, he thought that it would be a fun way to try it out. Overall, we had pretty split responses as to if this is something people would want to use in the future.
- <u>Feedback:</u>
    - When we explained our idea for a looper, everyone agreed and said that we should add it. Also, as mentioned before, highlighting pressed keys would make the program nicer. Someone, mentioned that they did not like how the black keys on the piano were so stubby, but later said it was actually easier to use. All in all, there was a consensus that our program is very user friendly and just a few additions would make it even better.

<u>**Refinement**</u>

After receiving our peer feedback, we decided that these are our next steps to create the final version of our program that we ourselves and our testers want to see happen:

- Have the keys be highlighted when pressed
    - We are currently working on the highlighted keys, it is just a long and tedious process.
- Looper
    - We figured out conceptually how to do the looper, we just need to code it and test it.
- More backtrack control
    - At the moment, we do have a working backtrack function, but we still want to figure out how to loop the bactrack so it plays over and over, and to have a backtrack automatically playing when the user opens the program.