Project Whitekey

Test Record

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

ID:CS338037 ID:CS338008 ID:CS338031

User Test (Belle Chasse High School band class)

Expected Input: An inexperienced user uses the keyboard to play along to a backing track (provided by youtube for the purposes of the test, due to this feature not being implemented at the time)

Expected Output: As long as the user presses keys with a solid rhythm, they will be able to create music that sounds correct.

Pass Criteria: A user, at random, will be able to play the instrument and have it sound pleasant

Fail Criteria: A user will either not be able to play the instrument or it will sound poor

Result: The keyboard performed as designed, and the users were pleased with its output.

Multitouch

Expected Input: the user presses multiple keys at a time

Expected Output: The sound of every pressed key is played simultaneously

Pass Criteria: Sounds should be able to play from many keys at once, and not be distorted when sounds are overlapping.

Fail Criteria: only one sound plays at a time, or the audio is distorted when multiple sounds play

Result: The output of the keyboard when using multitouch worked as designed, with multiple sounds playing simultaneously.

Audio Output

Expected Input: The user presses keys in an indeterminate order or combination with the goal of playing music.

Expected Output: Sounds should play which correspond to the key being pressed. Simultaneous key presses will result in simultaneous sounds.

Pass Criteria: The proper sound(s) plays immediately following a corresponding key press by the user.

Failure Criteria: If sounds do not play when a key (or simultaneous sounds when multiple keys are pressed) is pressed, the test will fail.

Result: The keyboard functioned as expected, producing the correct audio when they keys were pressed. Currently, some of the audio is a bit choppy. This will be fixed by re-cutting each audio file individually to ensure that they all sound well.

User test: Backing Track Functionality - Start

Expected Input: The user presses a play/stop button on the main interface.

Expected Output: A background track will play/stop at the current displayed BPM.

Pass Criteria: The music starts/stops as soon as the button is pushed.

Failure Criteria: If the background track does not play/stop when the start/stop button is pressed the test will fail.

Result: The Backing Track is currently unimplemented - It will be implemented in the final draft of the project for the demo.

User Test: Options Menu Button
Expected Input: The user presses an options button located on the main interface
Expected Output: An options menu will be displayed, allowing the user to select a different scale.
Pass Criteria: The options menu is displayed when the user presses the Options Menu button
Fail Criteria: The options menu is not displayed when the user presses the Options Menu button
Result: The Options Menu is currently unimplemented - It will be implemented in the final draft of the project for the demo

User Test: Initialization
Expected Input: The user opens the application.
Expected Output: The keys on the keyboard are mapped to the pentatonic scale
Pass Criteria: The app successfully loads and draws its graphical resources quickly and accurately. The app maps the pentatonic scale to the on-screen keys.
Failure Criteria: If the keys on the keyboard are mapped to something other than the pentatonic scale the test will fail.
Result: The app opened very quickly, and all keys were initially set to the pentatonic scale.

User Test: Application Exit Handling
Expected Input: The user closes out of the application.
Expected Output: Any currently playing sounds will be stopped.
Pass Criteria: The app is exited from and all resources being used by it are unloaded properly.
Failure Criteria: If sounds from the app are still being played after the user exits, the test will fail.
Result: The app stopped playing all sounds when the call to exit is given

User Test: Scale Select Option
Expected Input: The user selects a different scale for the keyboard in the options menu.
Expected Output: The keys to the keyboard should be remapped to play the respective sounds of the selected scale. When a piano key is pressed, it should reflect the respective element in the new scale. The current scale display will change to reflect the new scale.
Pass Criteria: The sounds are updated to reflect the key the music has been set to. The current scale display changes to reflect the new scale.
Failure Criteria: If the keys are not updated to play their new sound, or if the current scale display reads anything other than the new scale the test will fail.
Result: The Scale Select option is currently unimplemented - It will be implemented in the final draft of the project for the demo