Test Cases

Black-Box Test Cases

These test cases are defined based on the available documentation and execution of the program. The code was not inspected.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	Checking if the tones played are what they are told to be.	making of .wav files only with specific notes playing, and a way to check if the notes are exact.	playing of .wav file, and right after playing the specific note on an instrument	We expect the tones being exact, because they are exactly measured by hertz rate, and there's no more exact way of specifying tones	the observation passed, 8 out 8 tones were the same
2	Checking if the windows of swing function properly	-	testing all the windows	Other than the fact that not only the main windows exit button closes the app, but all existing windows (like the window of giving the name) that was already told by someone who implemented the app, everything should work as intended.	The observation passed.
3	Editing already put in lines	-	execute the app, put in some lines, and try to edit them	As a way to edit lines was only mentioned in the initial requirements	The observation passed.

				file, we dont expect editability	
4	Drawing lines	executing the app	draw in some lines	Smooth unproblematic making of the lines	The making of the lines is quite hard at times. Some clicks are not registered, or registered too late. After the fifth or so line, nothing happens on the first click, and multiple clicks are needed to properly put in a line
5	Naming, saving a file	executing the app	try to name a file, and save it	It should work as intended and save a file under the given name as a .wav	The observation passed
6	Playing of a file	executing the app	play an already existing file	Because we've heard it in the lecture, it should work	The observation passed
7					
8					

White-Box Test Cases

These additional test cases were defined during inspection of the code.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation (“pass” or failure description)
1	Constructor Test Line		construct the class Test with the implemented constructor then checking with input variables if anything goes wrong and calculating the length of the line then checking if the calculation in the constructor	The member variables of Line get initialized as wanted and length gets calculated correctly	pass
2	Constructor Test MyPoint		construct the class Test with the implemented constructor then checking with input variables if anything goes wrong	The member variables of MyPoint get initialized as wanted	pass
3	ToString Method		converting a MyPoint into a string via Method and checking with what should be printed	A given Tupel of points (one MyPoint) gets correctly converted to a String	pass
4	getDistance		Creating two MyPoints then calculating the distance between them, one time using the function one time using pythagoras, testing with various different inputs such as negative numbers, zero etc.	Function and Pythagoras formula outputs are the same	pass
5	roundPoint		creating two MyPoints then using the function on one of them in rounding/	points should have/shouldn’t have	pass

			not in rounding range and checking if the points are the same	the same value after function	
6	setY		Creating two MyPoints then setting the Y coordinate of one with setY and checking if it was set correctly	Y coordinate is set correctly	pass
7					
8					