

Studiengang Informatik (B.Sc.) WiSe 2022/23

Software Engineering 2

Übung 4

08. Dezember 2022

Ali Al Jasim (123741)
Daniel Ocks (124219)

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	Test constructor of Firefighter to check if the player created correctly	JUnit Test class	positive input, negative input	negative input for x, y coordinates and life points should not be valid	negative value for life points is valid (failure). others passed
2	Test the feature: take life points from the player	JUnit test class	tested with: positive input, negative input, zero input and a value bigger than creation input	negative or zero value should not affect life points. big values (bigger than input) should not be negative	for big values for taking life points the life points counter is negative (failure). others passed

Other features such as drawing objects, moving, shooting or enemy (abstract class) could not be tested with JUnit without inspecting the code. Thus we tested them by playing the game.

#	Test case (very brief description)	Expectation	observation (pass or failure description)
3	main feature: objects of the game should be shown	they should be shown since it is a main feature	pass
4	main feature: WASD key input for moving the player	it should work since it is a main feature	pass
5	main feature: shooting with left click on the mouse	it should work since it is a main feature	pass
6	extended feature: various kinds of enemies	it should work	pass
7	extended feature: enhanced interface	it should be easy to understand everything is shown	pass
8	extended feature: waves from enemies	it should work	pass
9	new features: various maps, goal of the game, restart game and enhanced movement of enemies	it should work	pass

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	key input class (WASD-input)	JUnit test class	testing if WASD-key are pressed, testing if they are released	movement variables should be changed	pass
2	testing all implemented features by inspecting the code				pass (everything is well implemented)
3	inspecting the code and reading every class and what it does			to find bugs	pass (there was no bugs in the written code)

Almost all implemented classes were hard to test with JUnit. Because almost every function depends on another function, so it was impossible to write test for a certain class without implementing other classes. the other reason is that the whole program is a graphic game which is not easy to test with JUnit. Thus we tested the classes and the features by reading and inspecting the code. The result is, we did not find bugs or wrong implementation for the described features in the implementation documents.