# Blackbox testing

## Purpose of Blackbox testing

Blackbox testing is a software testing technique which focuses on software behaviour and performance; the code base of a piece of software is not typically known and so the functionality of a given piece of software is determined to be working correctly, incorrectly or dysfunctional from a visual basis. This is measured from differences observed in the software’s output, from calling various functionality within the software, differences in the software’s output are subject to the working order of the software’s functionality.

Blackbox testing enables the identification of lacking and non-functioning engine functionality at a surface level; as there is no need to review the engines code base to acknowledge the implemented functionality’s working status. Being able to construct test cases quickly enables errors to be quickly acknowledged also; therefore, I have nominated this testing method for its suitability.

## Blackbox testing cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Case | Summary | Process | Actual result(s) | Expected result(s) | Passed? |
| 1 | Spdlog logger console window opens when application is launched | Launch application runtime event, observe for opening console window | Spdlog logger console window opens when application is launched | Spdlog logger console window opens when application is launched |  |
| 2 | Application window opens when application is launched | Launch application runtime event, observe for opening application window | Application window opens when application is launched | Application window opens when application is launched |  |
| 3 | Spdlog logger console window closes when application is terminated | Terminate application runtime event, observe for closing console window | Spdlog logger console window closes when application is terminated | Spdlog logger console window closes when application is terminated |  |
| 4 | Application window closes when application is terminated | Terminate application runtime event, observe for closing application window | Application window closes when application is terminated | Application window closes when application is terminated |  |
| 5 | Application window can be resized with mouse interaction | Click and drag mouse on edging perimeter of the application window, drag away from window | Application window is resized with mouse interaction | Application window is resized with mouse interaction |  |
| 6 | Application window can be minimized with mouse interaction | Click mouse upon minimize button on application window title bar | Application window is minimized with mouse interaction | Application window is minimized with mouse interaction |  |
| 7 | Application window can enter and exit windowed mode with mouse interaction | Click mouse upon windowed button on application window title bar | Application window is windowed and non-windowed with mouse interaction | Application window is windowed and non-windowed with mouse interaction |  |
| 8 | Application window can be terminated with mouse interaction | Click mouse upon exit button on application window title bar | Application window is terminated with mouse interaction | Application window is terminated with mouse interaction |  |
| 9 | Application window can be moved with mouse interaction | Click and drag mouse on application window title bar, drag in any direction and release mouse button | Application window is moveable with mouse interaction | Application window is moveable with mouse interaction |  |
| 10 | Application window can enter and exit full screen mode with button interaction | Press ‘SPACE’ button on keyboard, when application window is focused | Application window enters and exits full screen mode with button interaction | Application window can enter and exit full screen mode with button interaction |  |
| 11 | Application window can be terminated with button interaction | Press ‘ESC’ button on keyboard, when application window is focused | Application window terminates with button interaction | Application window terminates with button interaction |  |
| 12 | Application window loses focus when it is not the active window in use | Minimise the running application window or interact with another application program or interface | Application window loses focus when it is not the active window in use | Application window loses focus when it is not the active window in use |  |
| 13 | Application window becomes focused when it is the active window in use | Maximise the running application window and interact with the application window | Maximise the running application window and interact with the application window | Maximise the running application window and interact with the application window |  |
| 14 | Application window updates relative to vertical synchronisation | Enable vertical synchronisation window class variable, output the timestep in the logger console window | Application window updates relative to vertical synchronisation | Application window updates relative to vertical synchronisation |  |
| 15 | Application window draws and displays geometry | Launch application runtime event, observe for geometry in the form of cubes | Application window draws and displays geometry | Application window draws and displays geometry |  |
| 16 | Application window displays cubes direction of traversal to alternate every ‘10’ second interval | Launch application runtime event, observe for oscillating notion of movement in cube positions, observe logger console window for time passed | Application window displays cubes direction of traversal to alternate every ‘10’ second interval | Application window displays cubes direction of traversal to alternate every ‘10’ second interval |  |
| 17 | Application window displays textured phong cubes texture to switch when its direction of traversal alternates | Launch application runtime event, observe for textured phong cubes texture alternation when cubes direction of traversal alternates | Application window displays textured phong cubes texture to switch when its direction of traversal alternates | Application window displays textured phong cubes texture to switch when its direction of traversal alternates |  |
| 18 | Active camera object can be toggled with button interaction | Press ‘C’ button on keyboard, when application window is focused | Active camera object toggles between orthographic and perspective with button interaction | Active camera object toggles between orthographic and perspective with button interaction |  |
| 19 | Orthographic camera object can traverse upwards with button interaction | Press or hold ‘W’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object traverses upwards with button interaction | Orthographic camera object traverses upwards with button interaction |  |
| 20 | Orthographic camera object can traverse downwards with button interaction | Press or hold ‘S’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object traverses downwards with button interaction | Orthographic camera object traverses downwards with button interaction |  |
| 21 | Orthographic camera object can traverse left with button interaction | Press or hold ‘A’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object traverses left with button interaction | Orthographic camera object traverses left with button interaction |  |
| 22 | Orthographic camera object can traverse right with button interaction | Press or hold ‘D’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object traverses right with button interaction | Orthographic camera object traverses right with button interaction |  |
| 23 | Orthographic camera object can rotate clockwise with button interaction | Press or hold ‘E’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object rotates clockwise with button interaction | Orthographic camera object rotates clockwise with button interaction |  |
| 24 | Orthographic camera object can rotate anti-clockwise with button interaction | Press or hold ‘Q’ button on keyboard, when application window is focused, and orthographic camera is active | Orthographic camera object rotates anti-clockwise with button interaction | Orthographic camera object rotates anti-clockwise with button interaction |  |
| 25 | Perspective camera object can traverse forwards with button interaction | Press or hold ‘W’ button on keyboard, when application window is focused, and perspective camera is active | Perspective camera object traverses’ forwards with button interaction | Perspective camera object traverses’ forwards with button interaction |  |
| 26 | Perspective camera object can traverse backwards with button interaction | Press or hold ‘S’ button on keyboard, when application window is focused, and perspective camera is active | Perspective camera object traverses backwards with button interaction | Perspective camera object traverses backwards with button interaction |  |
| 27 | Perspective camera object can strafe left with button interaction | Press or hold ‘A’ button on keyboard, when application window is focused, and perspective camera is active | Perspective camera object strafes left with button interaction | Perspective camera object strafes right with button interaction |  |
| 28 | Perspective camera object can strafe right with button interaction | Press or hold ‘D’ button on keyboard, when application window is focused, and perspective camera is active | Perspective camera object strafes right with button interaction | Perspective camera object strafes right with button interaction |  |
| 29 | Perspective camera objects facing direction can be dynamically changed with mouse interactions | Click, drag and release left button on mouse, when application window is focused, and perspective camera is active | Perspective camera objects facing direction is dynamically changed with mouse interactions | Perspective camera objects facing direction is dynamically changed with mouse interactions |  |
| 30 | Backface culling render command is active and therefore culls back facing geometry | Launch application runtime event, control perspective camera to be positioned in front facing faces of any cube, observe distant world space | Backface culling render command is active, back facing geometry is not seen or rendered within camera viewport | Backface culling render command is active, back facing geometry is not seen or rendered within camera viewport |  |
| 31 | Depth test less render command is active and therefore enables character glyph to overlay cubes in the scene | Launch application runtime event, observe character glyph when positioned the same as a cube, relative to the active cameras viewport | Depth test less render command is active, character glyph renders and overlays cubes | Depth test less render command is active, character glyph renders and overlays cubes |  |
| 32 | Clear colour render command is active and therefore colours the application window that is backgrounding objects in the scene | Launch application runtime event, observe application window backgrounding colour | Clear colour render command is active and therefore colours the application window ‘grey’ | Clear colour render command is active and therefore colours the application window ‘grey’ |  |