Dynamic Analysis Report
CPE 353 - Software Design and Engineering
Team 13 – Castles & Creatures

Garett Sutton, Elijah Moore, Isaiah Denard, Alex Thomas

Gameserver

Class/Filename	# of Lines of Code	% Statement Coverage	Lines of Untested Code
gameserverdialog.cpp	301	88.04%	36
gameserverdialog.h	0	0	0
main.cpp	5	100%	0
mytcpserver.cpp	9	100%	0
mytcpserver.h	0	0	0
Totals	315	88.57%	36

Game_section_02

Class/Filename	# of Lines of Code	% Statement Coverage	Lines of Untested Code
defs.h	0	0	0
enemies.cpp	2	0	2
enemies.h	0	0	0
entityanimation.cpp	45	73.33%	12
entityanimation.h	0	0	0
main.cpp	5	100%	0
mainwindow.cpp	252	98.41%	4
mainwindow.h	0	0	0
player.cpp	67	86.57%	9
player.h	0	0	0
projectile.cpp	55	58%	23
projectile.h	0	0	0
sidescrollentity.cpp	194	94.85%	10
sidescrollentity.h	0	0	0
sidescrollscene.cpp	2	0	2

sidescrollscene.h	0	0	0
Totals	622	90.03%	62

Analysis of Results

Test coverage results summarized above are for the fully integrated system. Poor test coverage for file gameserverdialog.cpp are due to No instances where server was unable to start, Deconstructor is never called, No invalid connections were received, Database did not fail to open, no Errors executing statements, unimplemented functions, Debugging statements did not run. Poor test coverage for file entityanimation.cpp are due to the Debug Mode that boxes did not get run and unused constructor. Poor test coverage for file player.cpp are due to the unimplemented characters. Poor test coverage for file projectile.cpp are due to the code for changing bullet direction not used, but is implemented. Test coverage for file mainwindow.cpp not being 100% are due to an implemented but unused respawn button. Test coverage for file sidescrollentity.cpp not being 100% are due to deconstructors not being called and debug statements not being run and for a rare edge case when an entity receives a dead message twice. All .h files have 0 lines of executable code therefore no test coverage is necessary.