Anything added will be highlighted in yellow like so: EXAMPLE
Anything removed will be crossed out and highlighted in yellow like so: ~~EXAMPLE~~

Play Testing

Test ID	Description	Requirement ID	Description	Logic Test Result
1	GameShouldRunTest	SCR_RUNNABLE	The game should run without crashes	Pass
2	FireTruckShouldAttackIfInRangeTest	SFR_FORTRESS_DESTROY	Detecting an ET fortress in the firetruck's range should trigger the firetruck to start attacking it with a water jet	Pass
3	FortressShouldGetDestroyedTest	SFR_FORTRESS_DESTROY	After fatally damaging an ET fortress, it should be marked as 'destroyed'	Pass
4	FortressShouldAttackIfInRangeTest	SFR_FORTRESS_ATTACK UR_FORTRESS UR_FUN	Entering the range of an ET fortress should trigger the fortress to start attacking	Pass
5	FireTrucksShouldHaveDifferentStatsTest	UR_FIRETRUCKS_UNIQUE_SPEC	Each firetruck of the four should each have a specific statistic that differs it from the other three	Pass
6	ETShouldHaveUniqueSpecsTest	UR_ET_UNIQUE_SPEC	Each ET fortress should have unique statistics that make it different from other fortresses	~~Fail~~ Pass
7	TruckWaterTankShouldRefill	UR_FIRETRUCKS_REFILL, SFR_ALLOWED_TO_REFILL, SFR_CANCEL_REFILL, SFR_REFILL_OVER_TIME, SFR_REFILL_CONSTANT	Entering the range of the fire station should trigger the water refilling, assuming the water tank is not full	Pass
8	TruckHealthShouldRepairTest	UR_FIRETRUCK_REPAIR, SFR_ALLOWED_TO_REPAIR, SFR_CANCEL_REPAIR	Entering the range of the fire station should trigger the repairing, assuming the health bar is not full	Pass
9	ETPatrolsShouldDestroyFireStationTest	UR_ET_DESTROYS_STATION, UR_GAME_TIMER SFR_PATROL_FIRESTATION SFR_PATROL_DIFFICULTY	After 15 minutes of gameplay, the ET patrols should destroy the fire station	~~Fail - Not Implemented~~ Pass
10	GameShouldGetToGameOverScreenTest	UR_WIN_CONDITION, UR_LOSS_CONDITION SFR_ENDSCREEN	After destroying all ET fortresses or losing all four lives, the game should automatically reach the Game Over screen	Pass

11	GameShouldGetToGameOverScreenTest	SFR_MOVE_WHILE_DAMAGED	Getting hit by a bullet should not ~~empair~~ impair the truck's movement abilities	Pass
12	FireTruckShouldMoveWhileWaterTankEmptyTest	SFR_MOVE_WHILE_EMPTY	The fire truck should be able to move even when the water tank is empty	Pass
13	FireTruckShouldBeSelectedBeforeGameTest	SFR_FIRETRUCKS_STATS, SFR_FIRETRUCKS_SELECTION UR_FIRETRUCK_MIN_START	Before a new game is initiated, the user should be prompted with a fire truck selection screen	Pass
14	ScreenShouldSwitchTest	UR_MINIGAME, UR_DIFFICULTY_LEVEL, UR_CONTROLLER, UR_INSTRUCTIONS, UR_COLOUR_ACCESSIBILITY	The user should be able to move between ~~differnt~~ different screens without system bugs or crashes	Pass
15	FireTruckShouldNotDriveOnBuildingsTest	SFR_BUILDINGS UR_DRIVE	The firetruck should not be able to drive over buildings tiles	Pass
16	FireTruckShouldNotDriveOnRiversTest	SFR_RIVERS UR_DRIVE	The firetruck should not be able to drive over rivers tiles	Pass
17	HealthBarShouldAlwaysBeVisibleTest	SFR_HEALTH_BAR	The health bar should be visible at all point int time during gameplay	Pass
18	WaterBarShouldAlwaysBeVisibleTest	SFR_WATER_SUPPLY_BAR	The water bar should be visible at all point int time during gameplay	Pass
19	DifficultyHarder	SFR_PATROL_HEALTH SFR_PATROL_DIFFICULTY SFR_PATROL_DAMAGE UR_PATROL UR_ET_IMPROVEMENT	The game should become harder over time as the fortresses become more difficult to flood and the number of ET Patrols increase.	Pass
20	FortressNoChange	SFR_ET_LOCATIONS_NOT_CHANGEABLE UR_ET_MIN_START	The game should not allow the user to change locations of the fortresses.	Pass
21	LifeLevel	SFR_DESTROYED_TRUCKS	After a truck has been destroyed, there should be one less life on the heads-up display. This means the truck cannot be used again.	Pass
22	MiniGameOption	SFR_MINIGAME	On the main menu of the game you can click the minigame option and start playing the mini game.	Pass
23	ArrowControls	SFR_ARROWKEYS	The user should be able to move the fire truck with Arrow keys.	Pass
24	FlappyMiniGame	UR_MINIGAME	There should be a fully functional mini game based on flappy bird.	Pass

Junit tests

Fire Station Test (Run with JUnit FireStationTest)

Test ID	Test function name	Function tested	Function Use	Result of test	Test description
JUFS1	location()	getCentre()	Returns location of the fire station.	Pass	Checks if the location of the fire station is the correct location.
JUFS2	die()	die()	Kills the fire station.	Pass	Checks if the fire station can be destroyed.
JUFS3	update()	replenish()	Repairs the fire trucks health and refills its water supply.	Pass	Checks if fire station can repair and refill a fire truck.

Fire Truck Test (Run with JUnit FireTruckTest)

Test ID	Test function name	Function tested	Function Use	Result of test	Test description
JUFT1	Hitbox()	getHitbox()	Returns the hitbox of the fire truck.	Pass	This is a test to check if the hitbox generated is the right size.
JUFT2	movementTest()	getDirection()	Returns the direction the fire truck is facing.	Pass	This is a test to check if the directions of the fire truck work properly.
JUFT3	testInitialisation()	getHealthPoints()	Returns the health of the fire truck.	Pass	This is a test to see if the fire truck spawns with correct amount of health.
JUFT4	testRefill()	getHealthPoints() getCurrentWater()	Returns the health of the fire truck. Returns the water levels of the fire truck.	Pass	This is a test to see if the value of health and the value of water supply is correct after being repaired and refilled.

Fortress Test (Run with JUnit FortressTest)

Test ID	Test function name	Function tested	Function Use	Result of test	Test description
JUF1	takeDamage()	damage()	Lowers the health of a fortress by the amount within the brackets.	Pass	This test checks if damage to a fortress lowers the health by a correct amount.
JUF2	deathCheck()	death()	Removes a fortress from being active and	Pass	This test destroys a fortress and checks if it

			displays it as a destroyed state.		is displayed as a destroyed state.
JUF3	location()	getCentre()	Returns the location of the fortress.	Pass	This test checks if the fortress is in the right location.

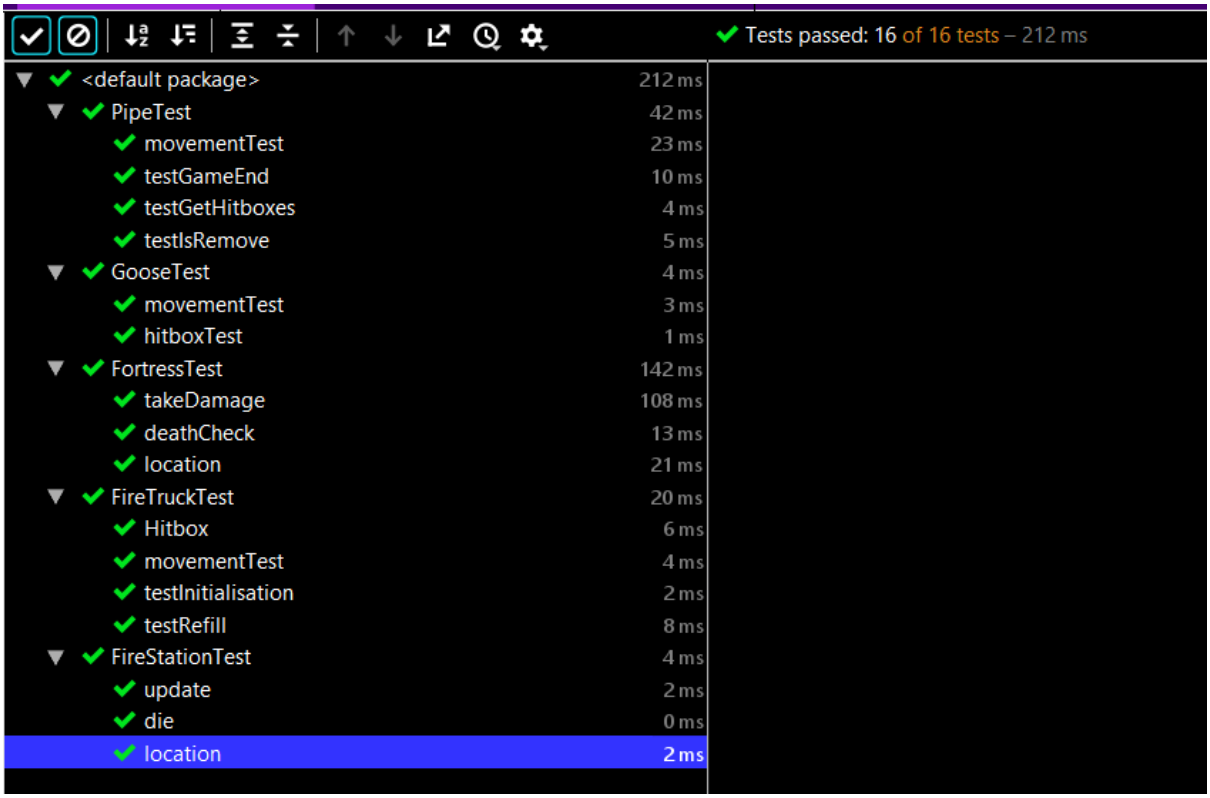
Goose Test (Run with JUnit GooseTest)

Test ID	Test function name	Function tested	Function Use	Result of test	Test description
JUG1	movementTest()	getY()	This returns the y value of the spaceship.	Pass	The first part of the test checks if the gravity works. The y value should become lower as time goes due to gravity. The second part of the test checks if the jumping function works, this time the y value should be greater to represent the spaceship going up.
JUG2	hitboxTest()	getHibox()	Returns the value of the hitbox.	Pass	This checks if the size of the hitbox generated is correct.

Pipe Test (Run with JUnit PipeTest)

Test ID	Test function name	Function tested	Function Use	Result of test	Test description
JUP1	movementTest()	getX()	This function returns the x value of the pipe.	Pass	This test is to check if the movement and the gravity works within the minigame.
JUP2	testIsRemove()	isRemove()	Returns true if the pipe can be removed.	Pass	This is a test to check if the pipe has been removed.
JUP3	testGetHitboxes()	getHitboxes()	Returns the hitboxes of the pipe.	Pass	This is a test to see if the hitbox is the correct size.
JUP4	testGameEnd()	gameEnd()	Returns whether the goose collides with the pipe.	Pass	This is a test to see whether the minigame has finished.

All the tests passing


Tests passed: 16 of 16 tests – 212 ms
<default package> 212 ms
PipeTest 42 ms
movementTest 23 ms
testGameEnd 10 ms
testGetHitboxes 4 ms
testIsRemove 5 ms
GooseTest 4 ms
movementTest 3 ms
hitboxTest 1 ms
FortressTest 142 ms
takeDamage 108 ms
deathCheck 13 ms
location 21 ms
FireTruckTest 20 ms
Hitbox 6 ms
movementTest 4 ms
testInitialisation 2 ms
testRefill 8 ms
FireStationTest 4 ms
update 2 ms
die 0 ms
location 2 ms

Acceptance Testing

TEST ID	REQUIREMENT ID	FIT CRITERION	RESULT	EVIDENCE
A_1	SNFR_INSTRUCTIONS	Instructions should cover all features of the game and how they work.	Pass	The game has a manual.
A_2	SNFR_TARGET_AUDIENCE	Game should be based on easy to understand rules, fast-paced and with relatively wide range of bullets' patterns difficulties	Pass	The game has simple controls, you only have to use the arrow keys to play the game. The map is simple and the shooting is automatic.
A_3	SNFR_JARGON	All user-facing messages shall be in plain English and will not use technical videogames jargon	Pass	The game tries to use the least amount of words as it possibly can so it is easy to understand.
A_4	SNFR_HIGHSCORES	The game should have a local record of the top high scores.	Pass	The game does not currently have a record of high scores.
A_5	SNFR_ACCESSIBILITY	There should be a way to modify the colour scheme in the for people who may be colour-blind.	Fail	No colour blind mode implemented
A_6	SNFR_MOBILE	The game should use an engine which allows you to easily transfer from pc to mobile.	Pass	The game is programmed on LIBGDX which can be easily transferred to android.

A_7	SNFR_TIME	You should be able to finish the game in under 5 minutes.	Pass	The game on average took 3 minutes to play for each user in our group.
A_8	SNFR_SIMPLE	The game should use arrow keys for the controls and the water cannons should be automatic.	Pass	The game uses up down left right arrow keys to control the game and if you are in close proximity to a fortress it will attack the fortress.
A_9	SNFR_FORTRESS	You are able to destroy all the fortresses in the game.	Pass	All the fortresses can be destroyed if the fire station is not destroyed.