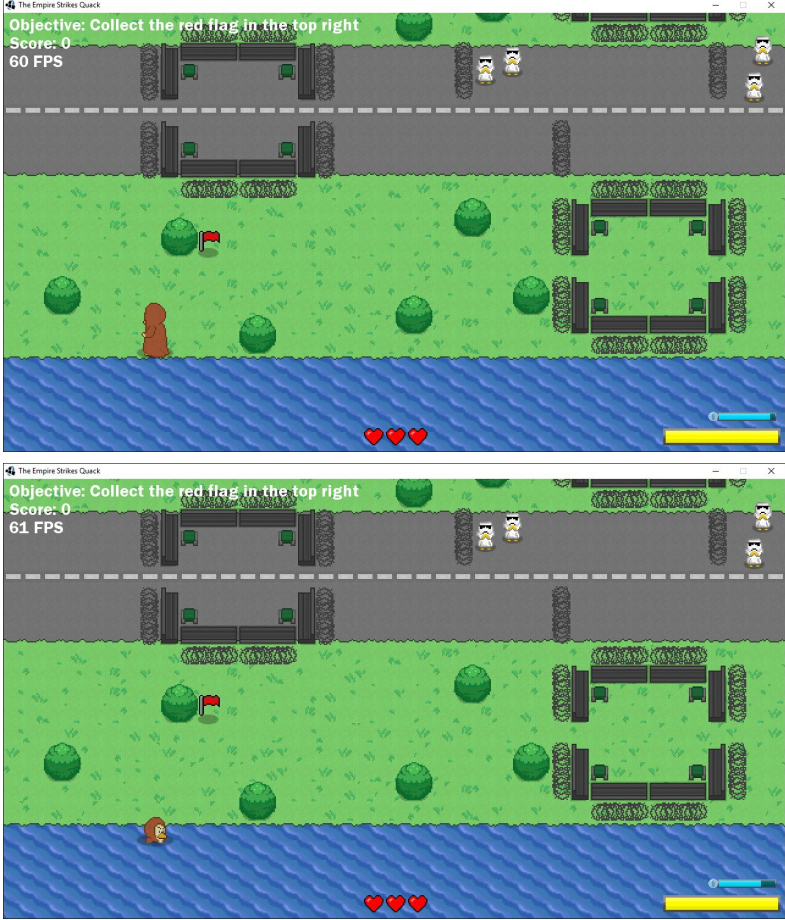


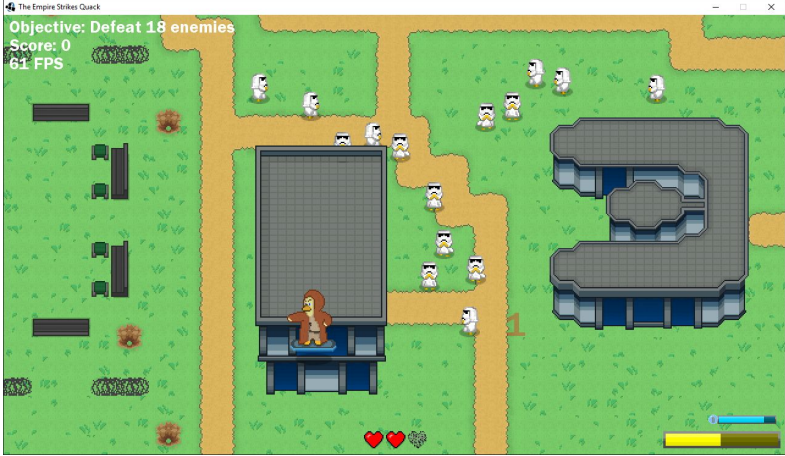



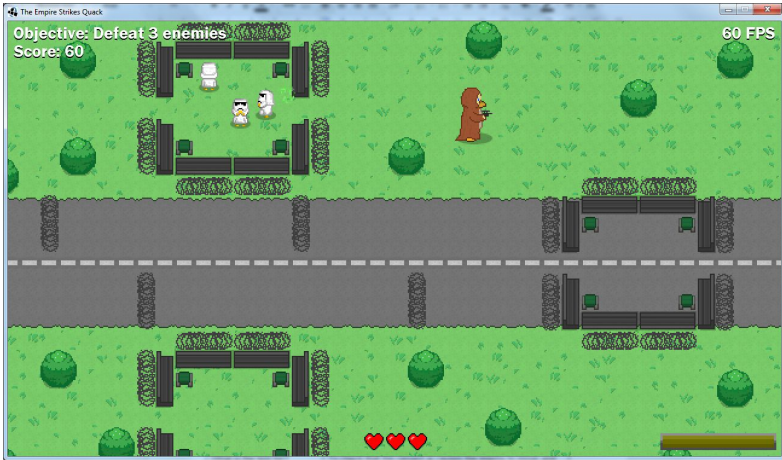
Part of game being tested	Flying System
Functions/Lines tested	player.update()
What is the test for?	To check after implementing flying that other base movement systems still work
How is the test being done?	Visual/Actively testing the game. Check player direction follows mouse still, moves correctly normally in water and on land. Check player transitions in and out of flying correctly
Test results	<p>Test was successful, Player still follows mouse correctly in land and on water and moves correctly with correct animations.</p> <p>Player enters flying correctly and can exit correctly with a button press.</p> 


	 <p>Objective: Collect the red flag in the top right Score: 0 61 FPS</p>
Actions to take	None


Part of game being tested	Flying System and collision
Functions/Lines tested	disableFlying() in player along with update() in player
What is the test for?	To check that the player does not exit flying when over a collision object that it would get stuck in
How is the test being done?	By testing in game, flying over an object and attempting to stop flying both by button-press and by letting the flying bar run out
Test results	<p>The player does not exit flying mode when over a collision object correctly.</p> <p>Retested after correction was attempted and test passed correctly.</p>  <p>Objective: Collect the red flag in the top right Score: 30 61 FPS</p>

	 <p>Objective: Defeat 18 enemies Score: 0 61 FPS</p>
Actions to take	None


Part of game being tested	Flying
Functions/Lines tested	player update function
What is the test for?	To check that the flight energy bar refills correctly
How is the test being done?	Fly over an obstacle while the bar has depleted and see what happens
Test results	<p>The bar seems to be depleting past 0% meaning it takes a very long time to refill</p>  <p>Objective: Defeat 3 enemies Score: 60 61 FPS</p>

	
Actions to take	<p>Prevent the flight energy from going below 0</p> <p>Retested after correction was attempted and test passed correctly.</p>


Part of game being tested	Ranged AI
Functions/Lines tested	Update and raycastProjectileCollides in RangedAI
What is the test for?	To check that the ranged ai causes the enemy to move and shoot in a satisfactory way. (aims towards the player, shoots at the correct regularity, moves and shoots properly)
How is the test being done?	Visually by moving around the enemy, moving behind obstacles and attempting movement that is likely to cause an issue to the pathfinding
Test results	<p>The enemy shot towards the player correctly, dealt with the player moving behind obstacles relatively well but sometimes would futilely continue shooting when the player was behind an object</p> 

	 <p>The Empire Strikes Quack Objective: Collect the red flag in the top right Score: 0 61 FPS</p> <p>The Empire Strikes Quack Objective: Collect the red flag in the top right Score: 0 61 FPS</p> <p>The Empire Strikes Quack Objective: Collect the red flag in the top right Score: 0 61 FPS</p>
Actions to take	<p>Further refinement of the parameters to improve the feel of the enemy movement.</p> <p>Values were tweaked which helped improve the feel. This is an iterative process though and can always be refined. This later gave the desired result.</p>

Part of game being tested	Ranged AI
Functions/Lines tested	RangedAI

What is the test for?	To check if mobs can hurt each other with their projectiles. This shouldn't be the case
How is the test being done?	Maneuvering into a position when a mob will shoot towards another mob and seeing the result
Test results	<p>The mobs can indeed damage each other which shouldn't happen</p> 
Actions to take	<p>An extra check will be added to the update function of the projectile class to make sure that mobs can't shoot each other.</p> <p>Retested after correction was attempted and test passed correctly.</p>

Part of game being tested	Player melee system
Functions/Lines tested	player update function and meleeupdate function
What is the test for?	to check if the player can use the melee attack when swimming. It shouldn't be able to
How is the test being done?	Moving into water and attempting to melee
Test results	The player can incorrectly melee when in water

	
Actions to take	<p>Prevent the player from meleeing when in water.</p> <p>Retested after correction was attempted and test passed correctly</p>

Part of game being tested	AngleTo function															
Functions/Lines tested	AngleTo function in entity <pre>public float angleTo(float x, float y) { return (float) Math.atan2(y - (this.y + this.getHeight()/2), x - (this.x + this.getWidth()/2)); }</pre>															
What is the test for?	To check that the values the function returns are sensible and correct															
How is the test being done?	Aiming the player in a particular direction and printing the angle returned from the function. Then comparing it against the expected range.															
Test results	<p>The output angle is in degrees, from +180 to -180</p> <p>The expected angle is a range to account for human error in aiming, however the range received reinforces that the function works correctly as it varies by a reasonable amount for small variances in the aiming direction</p> <table><tr><th>Direction</th><th>Expected(+/-2)</th><th>Actual</th></tr><tr><td>Right</td><td>0</td><td>0.65,0.38,-0.2</td></tr><tr><td>Up</td><td>90</td><td>90.8,90.9,89.2</td></tr><tr><td>Left</td><td>180,-180</td><td>179.5,-178.4,-178.9</td></tr><tr><td>Down</td><td>-90</td><td>-89.7,-89.1,-91.1</td></tr></table>	Direction	Expected(+/-2)	Actual	Right	0	0.65,0.38,-0.2	Up	90	90.8,90.9,89.2	Left	180,-180	179.5,-178.4,-178.9	Down	-90	-89.7,-89.1,-91.1
Direction	Expected(+/-2)	Actual														
Right	0	0.65,0.38,-0.2														
Up	90	90.8,90.9,89.2														
Left	180,-180	179.5,-178.4,-178.9														
Down	-90	-89.7,-89.1,-91.1														

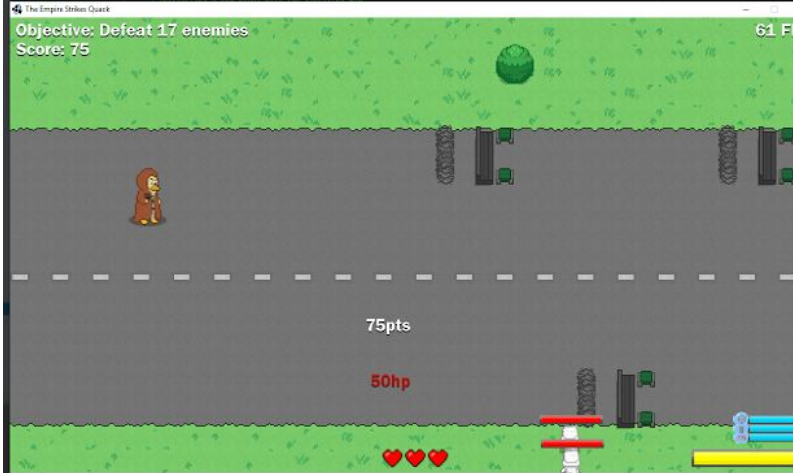
Actions to take	None, test succeeded

Part of game being tested	Ordering entities for rendering
Functions/Lines tested	GameScreen: Line 182
What is the test for?	Check that entities are rendered in the correct order depending on its type and its y position in the world.
How is the test being done?	<pre> public class EntitySortTest { public static List<Entity> entities; public static void main(String[] args) { entities = new ArrayList<>(); for (int x=0;x<4;x++) { for (int y=-1;y<2;y++) { switch (x) { case 0: entities.add(new Character(y)); break; case 1: entities.add(new Item(y)); break; case 2: entities.add(new Particle(y)); break; case 3: entities.add(new Projectile(y)); break; } } } int[][] comparisons = new int[entities.size()][entities.size()]; Entity.EntityComparator comparator = new Entity.EntityComparator(); for (int x=0;x<entities.size();x++) { for (int y=0;y<entities.size();y++) { System.out.printf("%2d ", comparator.compare(entities.get(x), entities.get(y))); comparisons[x][y] = comparator.compare(entities.get(x), entities.get(y)); } System.out.println(); } boolean symmetric = true; for (int x=0;x<entities.size();x++) { for (int y=0;y<entities.size();y++) { if (comparisons[x][y] != -comparisons[y][x]) symmetric = false; } } } } </pre>

	<pre> } } boolean transitive = true; for (int x=0;x<entities.size();x++) { for (int y=0;y<entities.size();y++) { for (int z=0;z<entities.size();z++) { if ((x != y x!= z) & y != z) { if (comparisons[y][x]==1 && comparisons[z][y]==1) { if (comparisons[z][x]!=1) { transitive = false; } } if (comparisons[y][x]==0 && comparisons[z][y]==0) { if (comparisons[z][x]!=0) { transitive = false; } } if (comparisons[y][x]==-1 && comparisons[z][y]==-1) { if (comparisons[z][x]!=-1) { transitive = false; } } } } } } System.out.println("SYMMETRIC: " + symmetric); System.out.println("TRANSITIVE: " + transitive); } } </pre>
Expected Result	<p>The contract for the comparator</p> <p>SYMMETRIC: true TRANSITIVE: true</p>
Test results	<pre> 0 1 1 0 1 1 0 0 0 0 0 0 -1 0 1 -1 0 1 0 0 0 0 0 0 -1 -1 0 -1 -1 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 -1 0 1 -1 0 1 -1 0 1 -1 0 1 -1 -1 0 -1 -1 0 -1 -1 0 -1 0 0 0 0 0 1 1 0 1 1 0 1 1 0 0 0 -1 0 1 -1 0 1 -1 0 1 0 0 0 -1 -1 0 -1 -1 0 -1 -1 0 0 0 0 0 1 1 0 1 1 0 1 1 0 0 0 -1 0 1 -1 0 1 -1 0 1 0 0 0 -1 -1 0 -1 -1 0 -1 -1 0 </pre> <p>SYMMETRIC: true TRANSITIVE: false</p>

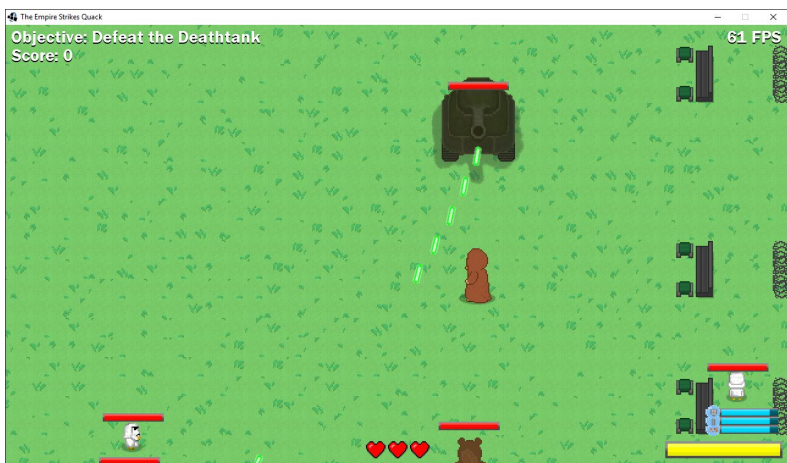
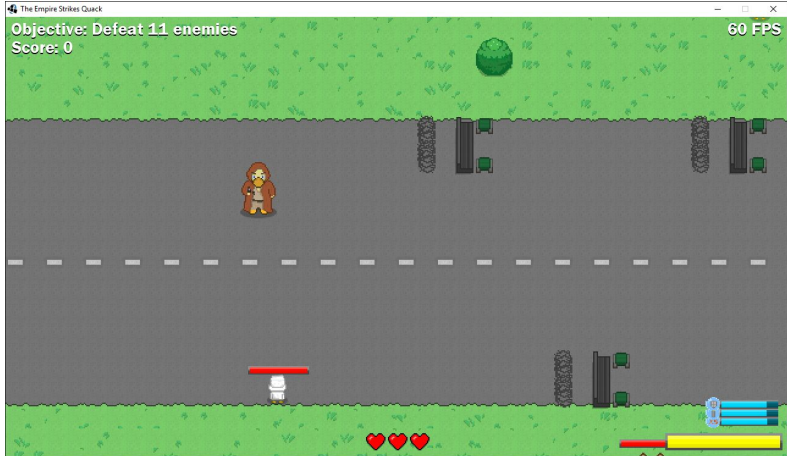
Actions to take	The testing shows that because particles should be rendered higher than all other entities, it breaks the requirement that comparisons must be transitive in java. Therefore as we require this functionality, we can handle the error in a try, catch block.
-----------------	--

Part of game being tested	Score system
Functions/Lines tested	Round update function
What is the test for?	To check that the score is correctly added when killing an enemy and that the numbers shown are accurate
How is the test being done?	<pre>int score = (int) (((Mob) entity).getScore()* (powerUpManager.getIsActive(PowerupManager.powerupTypes.SCORE_MULTIPLIER) ? Player.PLAYER_SCORE_MULTIPLIER : 1)); player.addScore(score); floatyNumbersManager.createScoreNumber(score, entity.getX(), entity.getY());</pre> <p>Kill various mobs, knowing what score they should give with and without the score multiplier and manually check the calculations</p>
Test results	<p>The stormtrooper ducks should give 15 points. Without the score multiplier (image 1), they are shown to give 15 points and the score correctly updates to show 15 points. With the score multiplier of 5 active (image 2) the same enemy correctly shows 75 and gives 75</p> 

	
Actions to take	None, test was successful

Part of game being tested	Enemy health bars
Functions/Lines tested	GameScreen update function
What is the test for?	To check enemy health bars are drawn correctly and update correctly
How is the test being done?	<pre> round.getEntities().stream().filter(entity -> entity instanceof Mob).forEach(entity -> { Mob chars = (Mob) entity; float offsetX = chars.getX() * 2 - chars.getWidth() / 2; float offsetY = chars.getY() * 2 + chars.getHeight() * 2; if (chars.getType() == Mob.MobType.BOSS) { offsetX += 40; offsetY += 15; } else if (chars.getType() == Mob.MobType.RANGED) { offsetX -= 7; offsetY += 75; } else { offsetX -= 17; offsetY += 10; } uiBatch2.draw(Assets.healthEmpty, offsetX, offsetY); Assets.healthFull.setRegionWidth((int) Math.max(0, ((float) chars.getCurrentHealth() / chars.getMaximumHealth()) * 100)); uiBatch2.draw(Assets.healthFull, offsetX, offsetY); }); </pre> <p>Attack enemies and view the health bars</p>
Test results	The stormtrooper duck has 100 max health and our shot does 50 damage. When shot the health bar correctly halved. The health bar also spawned in the correct place on the

stormtrooper duck, the squirrel and the boss.



Actions to take

None, test passed successfully