|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Gravity – sprite A at position B has gravity C applied to it for D time  Sprite Gravity – Sprite Ball at position 6m from bottom has gravity 9.8 applied to it for 5s | s: Ball  falling: 9.8ms^-2  mass: 3kg | Position:  Position:  S = ut+ ½(at)^2  S = ½(9.8\*5)^2  S = 1200.5m | Position: |  |
| 2 | Sprite Gravity – object A at position B has gravity C applied to it for D time  Sprite Gravity – Balloon at position 10m has gravity 6.3 applied to it for 10s time | s: Balloon  falling: 6.3ms^-2  mass: 0.5kg | Position:  S = ut+ ½(at)^2  S = ½(6.3\*10)^2  S = 1984.5m | Position: |  |

Work highlighted in **RED** is completed by Kanna

Work highlighted in **BLACK** is completed by Daniel.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Push – object A at position B and object C at position D collide at position E | s1:  s2:  mass1:  mass2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |
| 2 | Sprite Push – object A at position B and object C at position D collide at position E | s1:  s2:  mass1:  mass2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Physical Sprites Collision – object A at position B and object C at position D collide at position E | s1:  s2:  mass1:  mass2:  speed1:  speed2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |
| 2 | Physical Sprites Collision – object A at position B and object C at position D collide at position E | s1:  s2:  mass1:  mass2:  speed1:  speed2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Collision Angle – object A at position B and object C at position D collide at angle E | s1:  s2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |
| 2 | Collision Angle – object A at position B and object C at position D collide at angle E | s1:  s2: | Sprite 1 position:  Sprite 2 position: | Sprite 1 position:  Sprite 2 position: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Velocity after collied – object A at position B will collide with object C at position D. Velocity calculated after E time | sprite1: Soccer-ball  mass1: 3kg  velocity1: 5ms-1  sprite2: Basketball  mass2: 4kg  velocity2: -3ms^-1 | M1v1+M2V2 = M1v1f+M2V2f  3\*5+4\*-3 = 3\*v1f + 4\*v2f  V1f = 2.71ms-1  V2f = 4.71 ms-1  Sprite 1 velocity: 2.71ms-1  Sprite 2 velocity:  4.71ms-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |
| 2 | Velocity after collied – object A at position B will collide with object C at position D. Velocity calculated after E time | sprite1: Dodgeball  mass1: 7kg  velocity1: 3ms-1  sprite2: Bouncy Ball  mass2: 9kg  velocity2: 2ms^-1 | M1v1+M2V2 = M1v1f+M2V2f  7\*3+9\*-2 = 7\*v1f + 9\*v2f  Sprite 1 velocity:  1.875ms^-1  Sprite 2 velocity:  2.875ms^-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Kinetic Energy – object A will have B kinetic energy after C time | c1\_mass: 2kg  cl\_x\_speed: 3ms^-1  cl\_y\_speed: 4ms^-1 | Kinetic energy:  3^2+4^2=c^2  25 = c^2  C = 5  KE = 1/2\*m\*v^2  ½\*2\*5^2  =25J | Kinetic energy: |  |
| 2 | Kinetic Energy – object A will have B kinetic energy after C time | c1\_mass: 4kg  cl\_x\_speed: 1ms-1  cl\_y\_speed: 2ms-1 | Kinetic energy:  1^2+2^2=c^2  5 =c^2  C = 2.24ms-1  KE = ½\*m\*v^2  = 10.04J | Kinetic energy: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Velocity after Collied – object A will have B velocity after C time | c1\_mass: 5kg  c2\_mass: 4kg  cl\_x\_speed: 1ms^-1  cl\_y\_speed: 1ms^-1 | M1v1+M2V2 = M1v1f+M2V2f  5\*1+4\*1=5\*v1f+4v2f  Velocity\_1:  1ms-1  Velocity\_2:  1ms-1 | Velocity: |  |
| 2 | Velocity after Collied – object A will have B velocity after C time | c1\_mass: 2kg  c2\_mass: 1kg  cl\_x\_speed: 3ms-1  cl\_y\_speed: 4ms-1 | Velocity:  M1v1+M2V2 = M1v1f+M2V2f  2\*3+1\*4= 2v1f+1v2f  Velocity\_1 =  3.67ms^-1  Velocity\_2 =  2.67 | Velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Collision – circle A with B mass and C speed will have D speed after E time after collision | c1: Circle 1  c2: Circle 2  c1\_mass: 3kg  c2\_mass: 5kg  cl\_x\_speed: 2ms^-1  cl\_y\_speed: 10ms^-1 | Circle\_1= 12ms-1  Circcle\_2= 4ms-1 | Circle speed: |  |
| 2 | Collision – circle A with B mass and C speed will have D speed after E time after collision | c1: Circle 3  c2: Circle 4  c1\_mass: 5kg  c2\_mass: 8kg  cl\_x\_speed: 3ms-1  cl\_y\_speed: 6ms-1 | Circle speed:  Circle\_1 = 6.69ms^-1  Circle\_2 = 3.69ms^-1 | Circle speed: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Rect collision – rectangle A with B mass and C speed will have D speed after E time after collision | r1: Rectangle 1  r2: Rectangle 2  r1\_mass: 7kg  r2\_mass: 9kg  rl\_x\_speed: 5ms^-1  rl\_y\_speed: 6ms^-1 | Rectangle\_1 = 6.125ms^-1  Rectangle\_2 =  5.125ms^-1 | Rectangle speed: |  |
| 2 | Rect collision – rectangle A with B mass and C speed will have D speed after E time after collision | r1: Rectangle 1  r2: Rectangle 2  r1\_mass: 8kg  r2\_mass: 4kg  rl\_x\_speed: 8ms^-1  rl\_y\_speed: 10ms^-1 | Rectangle\_1 = 9.33ms^-1  Rectangle\_2 =  7.33ms^-1 | Rectangle speed: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Free Falling – object will fall, velocity is calculated | ratio: 2s  c1: Cup  Gravity: 9.8ms^-2 | Falling velocity:  V = gt  V = 9.8\*2  V = 19.6ms^-1 | Falling velocity: |  |
| 2 | Free Falling – object will fall, velocity is calculated | ratio: 4s  c1: Box  Gravity: 3.5ms^-2 | Falling velocity:  V = gt  V = 3.5\*4  V= 14ms^-1 | Falling velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Density – object’s density is calculated from volume | c1: Cup  c1\_mass = 5kg  volume = 4L | Density:  D = mass/volume  = 1.25kg/L | Density: |  |
| 2 | Density – object’s density is calculated from volume | c1: Mug  c1\_mass = 2kg  volume: 2L | Density:  D = mass/volume  D = 1kg/L | Density: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sinking Velocity – object’s sinking velocity is calculated | c1: 5kg  buoyancy: 2  volume: 3L  gravity: 9.8ms-2 | Sinking velocity:  F = s1mass\*s1speed  Density = s1mass/volume  F\_after\_water = square root ((F-buoyancy\*density) \*F buoyancy\*destiny)  V = F\_after\_water / s1mass  F = 5\*9.8 = 49  Density = 1.67  F\_after\_water = 86.44  V = 86.44/5  = 17.29 | Sinking velocity: |  |
| 2 | Sinking Velocity – object’s sinking velocity is calculated | c1:  4kg  buoyancy: 3  volume: 2L  gravity: 5.3ms^-2 | Sinking velocity:  F = s1mass\*s1speed  Density = s1mass/volume  F\_after\_water = square root ((F-buoyancy\*density) \*F buoyancy\*destiny)  V = F\_after\_water / s1mass  F = 4 \*5.3 = 21.2  Density = 2  F\_after\_water = 43.97  V = 43.97/4  = 10.99 | Sinking velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sinking – object A with B volume will be in C position after D time | c1:  Height\_of\_water:  buoyancy:  gravity:  volume:  density\_of\_water: | Position: | Position: |  |
| 2 | Sinking – object A with B volume will be in C position after D time | c1:  Height\_of\_water:  buoyancy:  gravity:  volume:  density\_of\_water: | Position: | Position: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Momentum X – object’s momentum is calculated from mass and velocity | Spr\_mass = 4kg  Spr\_velocity\_x = 3ms^-1 | Momentum X:  Spr\_mass\*spr\_velocity  4\*3= 12kms-1 | Momentum: |  |
| 2 | Sprite Momentum X – object’s momentum is calculated from mass and velocity | Spr\_mass = 5kg  Spr\_velocity\_x = 4ms^-1 | Momentum X:  Spr\_mass\*spr\_velocity  5\*4 = 20kgms^-1 | Momentum: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Momentum Y – object’s momentum is calculated from mass and velocity | Spr\_mass = 5kg  Spr\_velocity\_y = 6ms^-1 | Momentum Y:  Spr\_mass\*spr\_velocity  5\*6 = 30kgms^-1 | Momentum: |  |
| 2 | Sprite Momentum Y – object’s momentum is calculated from mass and velocity | Spr\_mass = 10kg  Spr\_velocity\_y = 6ms^-1 | Momentum Y:  Spr\_mass\*spr\_velocity  10\*6 = 60kgms^-1 | Momentum: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Inelastic Collision – sprite A with B mass and sprite C with D mass will have E and F velocity after inelastic collision, respectively | spr\_a: 12kg  spr\_b: 18kg  v\_a: 3ms^-1  v\_b: 0 ms^-1 | Sprite 1 velocity:  1.2ms^-1  Sprite 2 velocity:  1.2ms-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |
| 2 | Sprite Inelastic Collision – sprite A with B mass and sprite C with D mass will have E and F velocity after inelastic collision, respectively | spr\_a: 5kg  spr\_b: 10kg  v\_a: 5ms^-1  v\_b: 0ms^-1 | Sprite 1 velocity:  1.67ms^-1  Sprite 2 velocity:  1.67ms^-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Elastic Collision - sprite A with B mass and sprite C with D mass will have E and F velocity after elastic collision, respectively | spr\_a: 10kg  spr\_b: 23 kg  v\_a: 31ms^-1  v\_b 45ms^-1 | M1v1+M2V2 = M1v1f+M2V2f  Sprite 1  Velocity:  50.52ms^-1  Sprite 2  Velocity:  36.52ms^-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |
| 2 | Sprite Elastic Collision - sprite A with B mass and sprite C with D mass will have E and F velocity after elastic collision, respectively | spr\_a: 53kg  spr\_b: 41 kg  v\_a: 5ms^-1  v\_b 7ms^-1 | M1v1+M2V2 = M1v1f+M2V2f  Sprite 1  Velocity:  6.744ms^-1  Sprite 2  Velocity:  4.74ms^-1 | Sprite 1 velocity:  Sprite 2 velocity: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Confine Screen – sprite A will be within the screen | spr:  screen\_name: | Sprite within screen: | Sprite within screen: |  |
| 2 | Sprite Confine Screen – sprite A will be within the screen | spr:  screen\_name: | Sprite within screen: | Sprite within screen: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Impulse X – sprite’s impulse is calculated from mass and velocity | Spr\_mass: 4kg  Spr\_velocity: 9ms^-1 | Sprite impulse:  36kgms-1 | Sprite impulse: |  |
| 2 | Sprite Impulse X – sprite’s impulse is calculated from mass and velocity | Spr\_mass: 5kg  Spr\_velocity: 10ms^-1 | Sprite impulse:  50kgms-1 | Sprite impulse: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Impulse Y - sprite’s impulse is calculated from mass and velocity | Spr\_mass: 10kg  Spr\_velocity: 10ms^-1 | Sprite impulse:  100kgms-1 | Sprite impulse: |  |
| 2 | Sprite Impulse Y - sprite’s impulse is calculated from mass and velocity | Spr\_mass: 3kg  Spr\_velocity: 30ms^-1 | Sprite impulse:  90kgms^-1 | Sprite impulse: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Set Impulse X – sprite A will be at B position | Spr\_mass: 30kg  Spr\_impulse: 100kgms-1 | V=i/m  V = 3.33ms-1  Position: | Position: |  |
| 2 | Sprite Set Impulse X – sprite A will be at B position | Spr\_mass: 15kg  Spr\_impulse: 40kgms-1 | V=i/m  V = 2.67ms-1  Position: | Position: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Scenario | Input | Expected Result | Actual Result | Automated |
| 1 | Sprite Set Impulse Y - sprite A will be at B position | Spr\_mass: 15kg  Spr\_impulse: 43kgms-1 | V=i/m  V = 2.87ms-1  Position: | Position: |  |
| 2 | Sprite Set Impulse Y - sprite A will be at B position | Spr\_mass: 78kg  Spr\_impulse: 31kgms-1 | V=i/m  V = 0.4ms-1  Position: | Position: |  |