**Program’s expected behaviour**

Program’s basic action is a “roll” with value [0, 10]. Program’s larger units are “Frames”. Each frame must contain no less than 1 roll and no more than 3. Additionally, game has a value called “score”.

The game’s rules summarised:

1) If first rolls achieves 10 points (pins), it is called “Strike”. If roll is a “Strike” there can be no more rolls per this frame. User is given 10 score points plus sum of their next two rolls.

2) If last roll is a “Strike”, user is given 2 more attempts to roll. Rules of “Strike” applying to these text two rolls, but these two rolls cannot become “Strike” or “Spare” even if they meet other requirements of becoming them.

3) If user didn’t strike all pins at once, but did so in two rolls within a single frame, this action called “Spare”. For the “Spare” the user is awarded with additional score equal to the points they will achieve in the next roll.

4) If last two user’s rolls are a “Spare”, user allowed to make one more roll within same frame that cannot trigger “Strike” or “Spare”, but to which rules of “Spare” will apply

5) If no special conditions are met, user does 2 rolls per frame, sum of these rolls is added to the score.

**Unit test structure**

Parametrised unit test shall assept 3 values: “\_count\_numbers”, “expected”, “rolls”.

“\_count\_numbers” (int) – is a limit of how many “rolls” test should submit.

“expected” (int) – expected score to compare with.

“rolls” (list of integers) – bank of the rolls

**Test Cases**

Bank of the rolls:

rollsNormal = [10, 3, 6, 5, 5, 8, 10, 10, 10, 9, 0, 7, 3, 10, 10, 8]

rollsPerfect = [10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10]

rollsSpares = [5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]

rollsGutter = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

rollsRegular = [3, 4, 2, 5, 1, 6, 4, 2, 8, 1, 7, 1, 5, 3, 2, 3, 4, 3, 2, 6]

**Test case 1:** (count = 3, expected = 28, rolls = rollsNormal)

**Test case 2:** (count = 7, expected = 55, rolls = rollsNormal)

**Test case 3:** (count = 12, expected = 142, rolls = rollsNormal)

**Test case 4:** (count = 17, expected = 190, rolls = rollsNormal)

**Test case 5:** (count = 12, expected = 300, rolls = rollsPerfect)

**Test case 6:** (count = 21, expected = 150, rolls = rollsSpares)

**Test case 7:** (count = 20, expected = 0, rolls = rollsGutter)

**Test case 8:** (count = 20, expected = 72, rolls = rollsRegular)