Ten-pin Bowling Score Calculator – Coding Challenge v1.1

Implement a simple bowling score calculator, using the traditional scoring method specified here: https://en.wikipedia.org/wiki/Ten-pin_bowling#Traditional_scoring.

The input is the list of pins downed on each throw. A throw can have a value of 0 (no pins downed) to 10 (all pins down).

The output is the current score based on the pins thrown, the progress score of each frame already completed, as well as an indication as to whether the game is finished.

Note that as per the scoring explanation above, is possible that the progress score for a frame is not able to be determined, pending further throws. This happens in the frames directly after a spare or strike frame.

REST API

Implement the following API, a single REST endpoint that supports a single HTTP POST verb with the following contract:

POST /scores

Request payload
```
{
  "pinsDowned": [int]
}
```

Response
```
{
  "frameProgressScores": [string],
  "gameCompleted": boolean
}
```

If the progress score for a frame cannot be determined, it should be marked as "*".

Example 1 – Perfect Game

Request (in json)
```
{
  "pinsDowned": [10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10]
}
```

Response (in json)
```
{
  "frameProgressScores": [30, 60, 90, 120, 150, 180, 210, 240, 270, 300],
  "gameCompleted": true
}
```

Example 2 – 6 frames completed; all throws 1

Request (in json)

```
{
  "pinsDowned": [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
}
```

Response (in json)

```
{
  "frameProgressScores": [2, 4, 6, 8, 10, 12],
  "gameCompleted": false
}
```

Example 3 – 7 frames completed, spare and strikes example

Request (in json)

```
{
  "pinsDowned": [1, 1, 1, 1, 9, 1, 2, 8, 9, 1, 10, 10]
}
```

Response (in json)

```
{
  "frameProgressScores": [2, 4, 16, 35, 55, *, *],
  "gameCompleted": false
}
```

Submission

The solution needs to:

- be hosted online in some online GIT repository (GitHub, BitBucket or equivalent), that is publicly available (or made available on request).
- show the evolution of the solution. Progress commits are encouraged and expected.
- contain unit tests verifying logic and error conditions
- handle the following scenarios at least
  - ✓ perfect game
  - ✓ gutter-ball game

Some possibly useful information:

https://www.bowlinggenius.com/
https://www.liveabout.com/bowling-scoring-420895