

Oplæg til programmeringsopgave

Vi ønsker en algoritme/beregning af point der opnås i et spil Bowling.

Implementer en algoritme som beregner summen af opnåede points i et bowlingspil, der følger reglerne for 10-kegle spil.

Et spil består af 10 runder. I hver runde foretager spilleren 2 kast. Point tildeles afhængigt af antal væltede kegler.

Krav til løsningen

- Der skal beregnes point for den enkelte runde.
 - Der skal beregnes point for det enkelte spil (de 10 afviklede runder samlet set).
 - Opnåede points skal præsenteres visuelt på en selvvalgt facon.
 - Point tælles sammen efter den officielle metode *Traditional Scoring*.
 - Reglerne for *Strike* og *Spare* skal implementeres.
 - Unittest indgår som en del af den samlede kode.
 - Skitser en automatisk proces for build, test og deploy af din løsning.
- Der er fuld frihed til at vælge pseudo-kode til at beskrive dette.

Reglerne for *Traditional Scoring* kan findes på nettet. Kig eks. på https://en.wikipedia.org/wiki/Ten-pin_bowling#Traditional_scoring. Se i øvrigt referencer nedenfor.

Der er høj grad af frihed til løsning af opgaven og du er derfor frit stillet til at gribe opgaven an på den facon, du finder bedst egnet.

Den praktiske del

Indsendelse af løsning: Send enten alt relevant pr. mail eller et link til Github, hvis du benytter det:

- Henrik Schønning Henrik.Schoenning@ufst.dk

Hvis du vælger at sende din besvarelse pr. mail, skal du vide, at jeg tidligere har oplevet at zip-filer fanges i kontrol af indgående mail og automatisk fjernes, inden jeg ser dem.

Spørgsmål undervejs: Har du spørgsmål til opgaven eller andet, skal du ikke holde dig tilbage. Skriv til mig på ovenstående adresse.

Vægtning

Vi vægter selve implementeringen af pointberegningen og din tilgang dertil højst. Derefter implementering og tanker omkring unittest. På vores kommende møde vil vi derfor primært kigge på disse områder. Dette betyder, at en applikation der skriver til et konsolvindue vil være fuldt ud tilstrækkelig.

Har du lyst til, og erfaring med, at vise beregningen mv. på en anden visuel facon, gør du selvfølgelig bare det. Alternative metoder og gode ideer er altid velkomne.

Præsentation

På vores kommende møde forestiller vi os, at

- Du præsenterer din løsning og kode på en pc. Vi spørger og guider undervejs, hvis du bliver i tvivl om noget – det er vi meget øvede i. Vi lægger også gerne hardware til. Undervejs kan du f.eks. komme ind på flg.
 - Overordnet tilgang til løsning af opgaven.
 - Hvad var dine tanker om opgaven?
 - Hvilke klasser og metoder er anvendt?
 - Hvordan håndteres opnåede point for et givet spil?
 - Hvilke tanker havde du omkring design af din kode, når du samtidig skulle tænke unittest ind i løsningen?
- Vi lægger op til fælles snak om din tilgang til løsning af selve opgaven. Husk at snakken om løsningen er vigtigere end selve produktet.
- Skulle det ske, at du ikke kan få løsningen til at virke som du har planlagt, vil vi gerne understrege, at vi så blot tager udgangspunkt i det, når vi mødes. En sådan situation er jo noget vi som softwareudviklere ofte oplever som en del af udviklingsprocessen. Vi er vant til at hjælpe hinanden ud af situationer som denne, så det vil blot afspejle noget af vores hverdag.

Forudsætninger

- Programmeringssprog: Valgfrit. Vi foretrækker dog C# eller Java.
- Løsningsdesign/arkitektur: Valgfrit.
- UI: Valgfrit om nødvendig.
- Yderligere info
 - https://en.wikipedia.org/wiki/Ten-pin_bowling#Traditional_scoring
 - <https://da.wikipedia.org/wiki/Bowling>
 - Simulator: <http://www.bowlinggenius.com/>

Testdata

Ved eksekvering af dit program, kan du benytte visse af nedenstående 20 eksempler på komplette spil og fordeling af point. Bemærk at

- De 20 spil er indbyrdes uafhængige.
- Der er kun en enkelt aktiv spiller på banen.
- Alle tal henviser til point (antal væltede kegler) i de enkelte kast.
- Den enkelte linje indeholder antallet af væltede kegler i hvert af to kast i spillets samtlige 10 runder.
- Eksempler
 - I spil 1, runde 1 er der væltet 7 kegler i første kast og 1 kegle i andet kast. Der står 2 urørte kegler tilbage.
 - I spil 1, runde 2 vælter spilleren ikke en eneste kegle i rundens to kast. Der står 10 kegler tilbage.
 - I spil 1, sidste runde væltes 1 kegle i første kast og 6 kegler i andet kast. Der står således 3 urørte kegler tilbage på banen.
 - I spil 3 indleder spilleren med at lave en *strike*. Rundens kast nr. 2 foretages derfor ikke.

Data

Spil 1:

[7,1], [0,0], [6,2], [0,5], [4,4], [4,0], [7,2], [3,2], [5,5], [1,6]

Spil 2:

[5,4], [6,1], [0,0], [2,8], [7,3], [1,6], [5,2], [5,5], [4,0], [4,2]

Spil 3:

[10,0], [1,8], [6,3], [7,3], [6,2], [9,1], [2,2], [0,7], [5,3], [3,0]

Spil 4:

[0,5], [4,1], [5,1], [2,8], [3,5], [1,7], [4,4], [5,2], [4,3], [4,1]

Spil 5:

[2,7], [1,5], [7,2], [9,0], [3,4], [4,4], [9,0], [8,1], [6,4], [6,4,2]

Spil 6:

[10,0], [4,1], [6,0], [2,3], [2,5], [1,0], [7,1], [9,0], [5,4], [7,1]

Spil 7:

[0,3], [9,1], [1,1], [9,0], [10,0], [3,3], [0,9], [2,6], [2,5], [4,1]

Spil 8:

[8,2], [3,0], [7,0], [7,0], [0,3], [1,6], [2,5], [0,5], [2,7], [10,0,8]

Spil 9:

[2,6], [1,3], [2,6], [5,4], [4,3], [0,0], [1,2], [5,3], [6,3], [4,4]

Spil 10:

[9,0], [10,0], [0,7], [0,3], [8,0], [9,0], [5,5], [5,2], [4,2], [7,1]

Spil 11:

[5,1], [10,0], [10,0], [1,3], [4,3], [3,4], [2,1], [6,0], [0,2], [2,5]

Spil 12:

[8,1], [7,1], [7,2], [10,0], [1,0], [8,0], [6,1], [6,1], [2,1], [0,6]

Spil 13:

[1,7], [10,0], [8,0], [8,2], [4,5], [4,5], [8,0], [2,7], [10,0], [6,3]

Spil 14:

[4,3], [5,0], [5,2], [7,3], [5,4], [3,0], [10,0], [0,5], [0,5], [10,0,5]

Spil 15:

[2,4], [2,4], [6,2], [3,1], [5,3], [2,7], [8,0], [1,5], [9,0], [4,5]

Spil 16:

[1,2], [10,0], [5,3], [8,2], [2,4], [0,5], [5,0], [2,3], [0,5], [6,4,5]

Spil 17:

[10,0], [10,0], [1,9], [1,6], [10,0], [7,2], [9,0], [3,1], [9,0], [7,2]

Spil 18:

[9,1], [1,7], [6,4], [3,4], [2,4], [8,2], [7,2], [0,4], [4,1], [7,3,10]

Spil 19:

[4,1], [8,2], [2,8], [5,0], [10,0], [1,1], [7,1], [8,1], [8,1], [2,1]

Spil 20:

[9,1], [1,2], [8,0], [7,0], [2,3], [5,0], [8,0], [5,1], [2,7], [0,8]