Contents

Objective

Produce the score for one single-lane game of American ten-pin bowling
↳ Not checking if rolls are valid or invalid
↳ Not checking for the correct number of rolls and frames
↳ Not scoring intermediate frames (?)

Game Rules

<u>Ten-pin bowling:</u>
↳ Game consists of 10 "frames"
↳ Each frame provides players two opportunities to knock down all ten pins
↳ The score for a frame is the total number of pins knocked down, and any bonuses
↳ A "spare" is when all 10 pins are knocked down in those two attempts, thus completing the frame
    ↳ Bonus: the first roll of the next frame; that score will therefore count twice!
↳ A "strike" is when all 10 pins are knocked down in the very first roll of a frame, essentially completing the frame
    ↳ Bonus: the score of the next two rolls!
↳ This way, the max score for a frame is 30, including bonuses
    ↳ Thus for a game of 10 frames, it's 300
↳ Additionally, if a player scores a strike or a spare in the final (10th) frame, they do get to roll the extra balls but no more than three rolls are allowed in the final frame, this way a game can have a maximum of 12 strikes

Approach

1. The core-nature:
   → A roll can be valid or invalid, but since that's out of scope, let's assume all rolls are valid
   → A valid roll may make contact with the pins, or it may go to the gutter
   → If it does strike the pins, some or all may fall over:
   ↳ Precisely calculating which pins fall will require a physics engine, this brings us to an important fork in our design approach!
   → The game can be designed to be luck-based or skill-based:
   ↳ A skill-based game will require physics calculation determining the roll-path of the ball and the pin-impact
   ↳ A luck-based game will simply proceed by randomly picking outcomes, we start with this approach!

2. A luck-based approach:
   → Central to this approach would be a random Boolean: TRUE(1) for a favorable outcome, FALSE(0) otherwise
   → A roll strikes the pins (Boolean: TRUE (1)) or misses and goes to the gutter (Boolean: FALSE(0))
   → Since this is luck-based and not skill/physics-based, let's assume that if the pins are struck by our ball, they collide either head-on or in the pockets:
   ↳ Statistically, the pockets are more favorable for a strike, so let's assume that hitting the pocket gets us a strike
   ↳ Conversely, hitting head-on is more unpredictable, so let's assume a head-on hit can lead to either a strike where all the pins fall or we end up with a split where some are still left standing
   ↳ If it's a split, the remaining roll will either knock down the remaining pins or we're left with an open frame!

Pseudo-Code

Calculate GameScore()

C:\Users\abhee\OneDrive\Documents\Study\Assignment - Take Home\pseudo.py - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

pseudo.py

```python
def calculate gamescore():

    frame_scores = {}  #dict_containing roll scores as FRAME_NO:<ROLL_1><ROLL_2>
    GAME_SCORE = 0

    for i in range(1, len(frame_scores)):
        FRAME_SCORE = 0

        #Intuition: The score will only be '10' for strikes, else it'll be a list of roll scores

        if strike:

            #calculate bonus = next two rolls:

            if (next frame also strike):

                if (second roll also strike):
                    #calculate score

                else:
                    bonus = [strike_frame_i+1] + [roll_1_of_frame_i+2]

            else (next frame not strike):
                bonus = [roll_1_of_frame_i+1] + [roll_2_of_frame_i+1]

            calculate FRAME_SCORE

        elif not strike:

            score = [roll_1_of_frame_i] + [roll_2_of_frame_i]

            if score == 10:
                Spare!
                bonus = [next_roll_score]

            calculate FRAME_SCORE

        update GAME_SCORE

    return GAME_SCORE
```

GameMode()

```
def gameMode():

    FINAl_STRIKE = False
    FINAL_SPARE = False

    for i in range(1,12):       #Frame 11 (bonus_frame) can hold two bonus rolls

        if bonus_frame and not FINAL_STRIKE and not FINAL_SPARE:
            frame_scores.update({11:[0]})
            continue
        elif bonus_frame and FINAL_STRIKE:
            score = []
            for j in range(2):
                bonus_roll()
                update score
            frame_scores.update({11:score})
            continue
        elif bonus_frame and FINAL_SPARE:
            score = bonus_roll()
            frame_scores.update({11:score})
            continue


        current_framescore = 0
        WAS_STRIKE = False
        WAS_SPARE = False

        for j in range(2):      #two rolls a frame, unless strike

            if roll_2 and WAS_STRIKE:
                continue        #progress to next frame

            user_input = enter_to_roll

            if pocket_hit:      #consider strike
                frame_scores.update({10})
                WAS_STRIKE = True

            elif head-on:       #check if strike or not

                if strike:
                    frame_scores.update({10})
                    WAS_STRIKE = True

                elif not_strike:

                    if roll_1:
                        current_framescore += calculate_score_1_to_9

                    if roll_2:
                        calculate_score = random(total_pins - pins_standing_after_roll_1)
                        update current_framescore

                        if Spare:
                            WAS_SPARE = True

            if final_frame:
                if WAS_STRIKE:
                    FINAl_STRIKE = True
                elif WAS_SPARE:
                    FINAL_SPARE = True

    calculate_gamescore()
    print("congrats!")
```

Main() and misc. functions

```
def favorable():
    outcome = random.choice([True, False])
    return outcome


def ascii_art_functions():

    print()
    print()
    print()



frame_scores = {}

def main():

    user_input = gameMode or testMode

    if gameMode:
        user_input = disable_ASCII_Art?
        if disable:
            DISABLE_ASCII = True

        gameMode()

    elif testMode:
        define_input_regex
        print_guidance

        frame_counter = 1
        while True:
            try:
                user_input
                if not regex_match:
                    raise ValueError
            except ValueError:
                print("Please re-enter")
                continue

        frame_scores.update({})
        if frame_counter == 12:
            calculate_gamescore()
            break
```

Random Booleans

Two common ways to generate random Booleans in Python, demonstrated and timed below:

```
Windows PowerShell (x86)    Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\abhee> python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import random
>>> import timeit
>>>
>>> def bool_gen_appr1():
...     randBool = random.choice([True, False])
...
>>> def bool_gen_appr2():
...     randBool2 = bool(random.randint(0,1))
...
>>> timeit.timeit(bool_gen_appr1)
0.2740244999999959
>>>
>>> timeit.timeit(bool_gen_appr2)
0.46009200000000305
>>>
>>>
```

For obvious reasons, approach 1 is better so proceeding with that

Execution Instructions

Simply execute “python ten_pin.py” on CLI:

```
Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\abhee\OneDrive\Documents\Study\Assignment - Take Home> python .\ten_pin.py
Select [t]est mode or [g]ame mode:
```

Docker Deployment:

Makefile:

```
FROM python:3.9-slim-buster
RUN useradd --create-home --shell /bin/bash app_user
WORKDIR /home/app_user
USER app_user
COPY . .
CMD ["bash"]
CMD [ "python", "./ten_pin.py" ]
```

Build & Run the Container:

docker build -t tenpin_container .

docker run -it --name tenpin tenpin_container

```
Windows PowerShell
PS C:\Users\abhee\OneDrive\Documents\Study\Assignment - Take Home> docker build -t tenpin_container .
[+] Building 0.3s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                    0.0s
 => => transferring dockerfile: 32B                                                                     0.0s
 => [internal] load .dockerignore                                                                       0.0s
 => => transferring context: 2B                                                                         0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim-buster                               0.2s
 => [1/4] FROM docker.io/library/python:3.9-slim-buster@sha256:0ac1e4889c90d3117680b33cbdd397e3d95098b4c6bcff8600  0.0s
 => [internal] load build context                                                                       0.0s
 => => transferring context: 789B                                                                       0.0s
 => CACHED [2/4] RUN useradd --create-home --shell /bin/bash app_user                                   0.0s
 => CACHED [3/4] WORKDIR /home/app_user                                                                 0.0s
 => [4/4] COPY . .                                                                                      0.0s
 => exporting to image                                                                                  0.0s
 => => exporting layers                                                                                 0.0s
 => => writing image sha256:c7e7288c107be0461f8b7d1cda081a8d9ba472b0290b038f34da94ff0b2ddb67             0.0s
 => => naming to docker.io/library/tenpin_container                                                     0.0s
PS C:\Users\abhee\OneDrive\Documents\Study\Assignment - Take Home> docker run -it --name tenpin tenpin_container
Select [t]est mode or [g]ame mode:
```

Unit Tests

Perfect Game

```
Windows PowerShell

Roll scores: 10
Frame  2 :
Roll scores: 10
Frame  3 :
Roll scores: 10
Frame  4 :
Roll scores: 10
Frame  5 :
Roll scores: 10
Frame  6 :
Roll scores: 10
Frame  7 :
Roll scores: 10
Frame  8 :
Roll scores: 10
Frame  9 :
Roll scores: 10
Frame  10 :
Roll scores: 10
Frame  11 :
Roll scores: 10,10

Final Scores:  {1: 10, 2: 10, 3: 10, 4: 10, 5: 10, 6: 10, 7: 10, 8: 10, 9: 10, 10: 10, 11: [10, 10]}

Frame score for frame  1  is:  30
Frame score for frame  2  is:  30
Frame score for frame  3  is:  30
Frame score for frame  4  is:  30
Frame score for frame  5  is:  30
Frame score for frame  6  is:  30
Frame score for frame  7  is:  30
Frame score for frame  8  is:  30
Frame score for frame  9  is:  30
Frame score for frame  10  is:  30
Final Game Score is:  300
Select to [r]estart or [e]xit:
```

Last Frame Spare

```
Frame  1 :
Roll scores: 10
Frame  2 :
Roll scores: 10
Frame  3 :
Roll scores: 10
Frame  4 :
Roll scores: 10
Frame  5 :
Roll scores: 10
Frame  6 :
Roll scores: 10
Frame  7 :
Roll scores: 10
Frame  8 :
Roll scores: 10
Frame  9 :
Roll scores: 10
Frame  10 :
Roll scores: 7,3
Frame  11 :
Roll scores: 10

Final Scores:  {1: 10, 2: 10, 3: 10, 4: 10, 5: 10, 6: 10, 7: 10, 8: 10, 9: 10, 10: [7, 3], 11: 10}

Frame score for frame  1  is:  30
Frame score for frame  2  is:  30
Frame score for frame  3  is:  30
Frame score for frame  4  is:  30
Frame score for frame  5  is:  30
Frame score for frame  6  is:  30
Frame score for frame  7  is:  30
Frame score for frame  8  is:  27
Frame score for frame  9  is:  20
Frame score for frame  10  is:  20
Final Game Score is:  277
Select to [r]estart or [e]xit:
```

Invalid Input – Exception Handling

```
Frame  1 :
Roll scores: 10
Frame  2 :
Roll scores: 11
Invalid input: enter 10 for strikes or comma-separated roll-scores without spaces for the frame
Frame  2 :
Roll scores: 10
Frame  3 :
Roll scores: 3,0
Frame  4 :
Roll scores: 4,1
Frame  5 :
Roll scores: 6,
Invalid input: enter 10 for strikes or comma-separated roll-scores without spaces for the frame
Frame  5 :
Roll scores: 6,0
Frame  6 :
Roll scores: 10
Frame  7 :
Roll scores: 10
Frame  8 :
Roll scores: 0
Invalid input: enter 10 for strikes or comma-separated roll-scores without spaces for the frame
Frame  8 :
Roll scores: 0
Invalid input: enter 10 for strikes or comma-separated roll-scores without spaces for the frame
Frame  8 :
Roll scores: 0,0
Frame  9 :
Roll scores: 0,0
Frame  10 :
Roll scores: 0,0
Frame  11 :
Roll scores: 0,0

Final Scores:  {1: 10, 2: 10, 3: [3, 0], 4: [4, 1], 5: [6, 0], 6: 10, 7: 10, 8: [0, 0], 9: [0, 0], 10: [0, 0], 11: [0, 0]}

Frame score for frame  1  is:  23
Frame score for frame  2  is:  13
Frame score for frame  3  is:  3
Frame score for frame  4  is:  5
Frame score for frame  5  is:  6
Frame score for frame  6  is:  20
Frame score for frame  7  is:  10
Frame score for frame  8  is:  0
Frame score for frame  9  is:  0
Frame score for frame  10  is:  0
Final Game Score is:  80
Select to [r]estart or [e]xit:
```

Random Inputs With Spare Frame

```
PS C:\Users\abhee\OneDrive\Documents\Study\Assignment - Take Home> python .\ten_pin.py
Select [t]est mode or [g]ame mode: t
Enter roll scores (not frame scores) in the following format:
Strike Frame: 10
Spare Frame: 3, 7
Open Frame w/ Gutter: 5, 0
One Bonus roll: 7
Two Bonus rolls: 10, 10
No bonus rolls: 0


Frame  1 :
Roll scores: 3,2
Frame  2 :
Roll scores: 4,5
Frame  3 :
Roll scores: 6,1
Frame  4 :
Roll scores: 0,0
Frame  5 :
Roll scores: 0,1
Frame  6 :
Roll scores: 0,4
Frame  7 :
Roll scores: 5,0
Frame  8 :
Roll scores: 7,3
Frame  9 :
Roll scores: 6,4
Frame  10 :
Roll scores: 9,0
Frame  11 :
Roll scores: 0,0

Final Scores:  {1: [3, 2], 2: [4, 5], 3: [6, 1], 4: [0, 0], 5: [0, 1], 6: [0, 4], 7: [5, 0], 8: [7, 3], 9: [6, 4], 10: [9, 0], 11
: [0, 0]}

Frame score for frame  1  is:  5
Frame score for frame  2  is:  9
Frame score for frame  3  is:  7
Frame score for frame  4  is:  0
Frame score for frame  5  is:  1
Frame score for frame  6  is:  4
Frame score for frame  7  is:  5
Frame score for frame  8  is:  16
Frame score for frame  9  is:  19
Frame score for frame  10  is:  9
Final Game Score is:  75
Select to [r]estart or [e]xit:
```

GameMode - With ASCII Art

```
Select to [r]estart or [e]xit: r
Select [t]est mode or [g]ame mode: g
Disable ASCII Art? [y]es or [n]o: n

FRAME  1

Press Enter to roll...

==========================================================================================
                  >                                                                     |
                    >                                                                 |
                       >                                                            |   |
 O                        >                                                        |   |
                       >                                                            |   |
                    >                                                                 |
                  >                                                                     |
==========================================================================================



==========================================================================================
                  >                                                                     |
                    >                                                                 |
                       >                                                            |   |
                          >    O                                                     |   |
                       >                                                            |   |
                    >                                                                 |
                  >                                                                     |
==========================================================================================



==========================================================================================
                  >                                                                     |
                    >                                                                 |
                       >                                                            |   |
                          >                                         O                |   |
                       >                                                            |   |
                    >                                                                 |
                  >                                                                     |
==========================================================================================



ROLL  1
HEAD-ON STRIKE!!

==========================================================================================
                  >
                    >
                       >
                          >                                                  STRIKE!!
                       >
                    >
                  >
==========================================================================================



roll score: 10


FRAME  2

Press Enter to roll...
```

Index

GameMode – Without Art

```
ROLL  1
STRIKE!!
roll score: 10


FRAME  9

Press Enter to roll...

ROLL  1
HEAD-ON STRIKE!!
roll score: 10


FRAME  10

Press Enter to roll...

ROLL  1
STRIKE!!
roll score: 10


Press Enter to roll...

You rolled into the gutter!
roll score:  0

Press Enter to roll...

STRIKE!!
roll score:  10

Final Scores:  {1: 10, 2: 10, 3: [5, 3], 4: 10, 5: [0, 6], 6: 10, 7: 10, 8: 10, 9: 10, 10: 10, 11: [0, 10]}

Frame score for frame  1  is:  25
Frame score for frame  2  is:  18
Frame score for frame  3  is:  8
Frame score for frame  4  is:  16
Frame score for frame  5  is:  6
Frame score for frame  6  is:  30
Frame score for frame  7  is:  30
Frame score for frame  8  is:  30
Frame score for frame  9  is:  20
Frame score for frame  10  is:  20
Final Game Score is:  203
CONGRATULATIONS!
Select to [r]estart or [e]xit:
```