

1619313040/3301 (Data Science Oriented Programming Language-Python)

SEMESTER 2 (Spring), 2021/2022

LABORATORY WORK ONE

NAME: MD Omar Faruq

STUDENT NO.: 201939060073

SUBMITTED TO:

Assoc. Prof. FANG YU

MARKING SCHEME (TO BE FILLED BY THE LECTURER)

|  |  |
| --- | --- |
| **CRITERIA** | **MARKS** |
| CONTENTS | /6 |
| ORGANIZATION/STRUCTURE | /2 |
| WRITING MECHANICS | /2 |
| **TOTAL** | **/10** |

**PYTHON BASIC SYNTAX PRACTICE**

In this section, students must display originality of their writings by jotting down their understanding of Python programming language. Students need to fully describe and discuss in terms of:

1. **The flowcharts for every version of your code block**
2. **Screenshot your running results (better applying video clips), upload your source codes and your lab work report.**

*FOLLOWING IS THE LABWORKS (INCLUDING 3 VERSIONS OF THE GAME)*

**Description about the task:**

Developing a little fighting game, in this game, the attributes of single player and PC player are generated randomly. The players are fighting until one counterpart’s blood bar (Health Point) is < 0

This kind of fighting will take 3 rounds. The winner is the best of 3 sets. Finally, print out the results and show ‘WHO IS THE WINNER’.

Three Versions of the Game

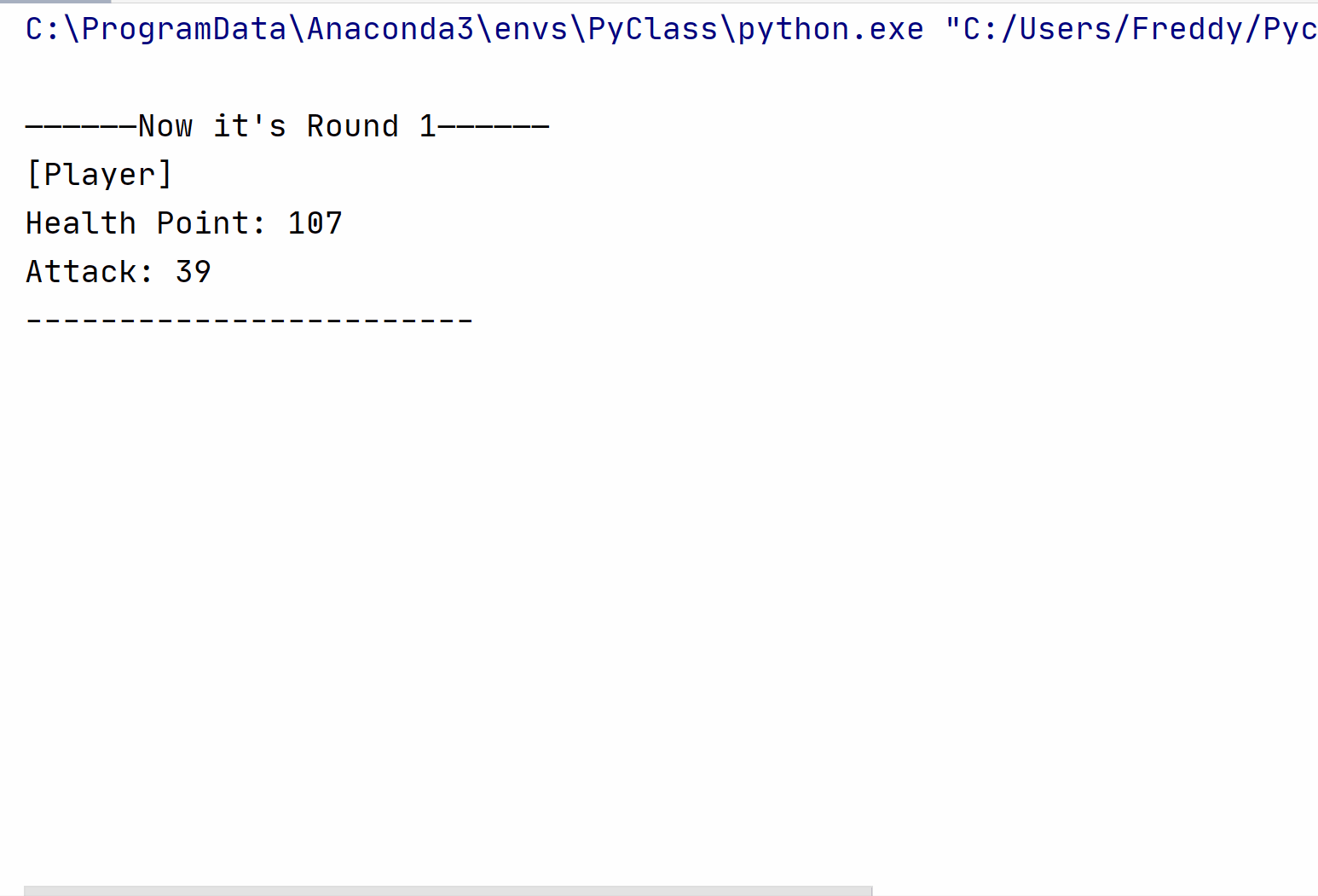
**Version 1.0**: setup the attributes of players, computing the blood bar manually when fighting and print out the combat course (fighting progressing)

**Version 2.0**: generate the attributes value of players randomly, computing the blood bar automatically when fighting and optimizing the ‘displaying’ of the combat course.

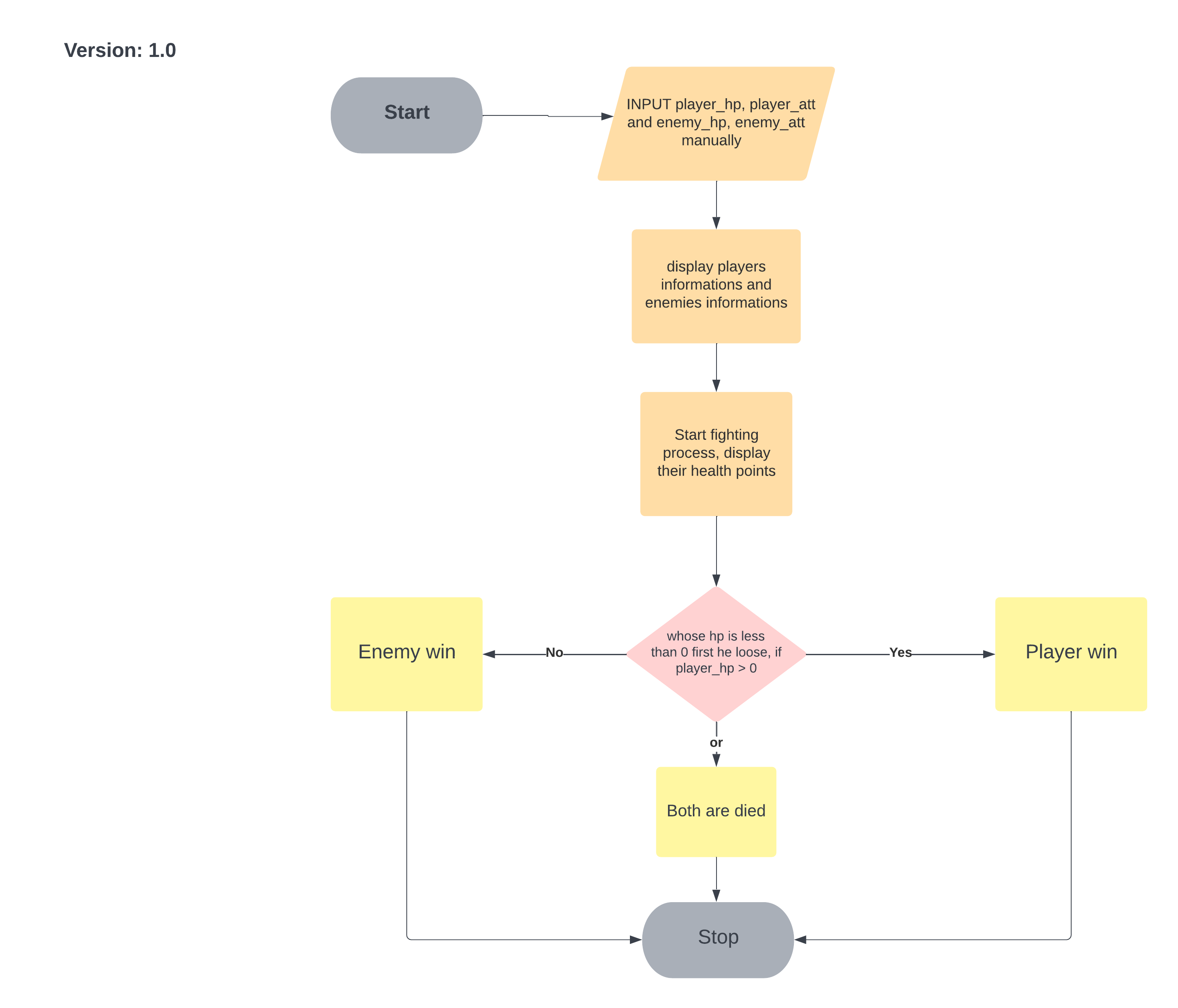
**Version 3.0**: 3 round PK, display the winner and loser each round. The best of 3 sets, present who is the finally winner and loser.

Feel the procedure of ‘How to solve problem’, “decompose the problem🡪execute🡪get another problem🡪continue executing” the iterative process.

Here is the demo for the Version 3’s output



**Flowchart for version 1.0**



**Source code for version 1.0**

import time  
  
print('[Player]')  
player\_hp = 150  
player\_att = 35  
print(f'Heath Point: {player\_hp}')  
print(f'Attack: {player\_att}')  
print('-------------------------------------------------------------------\n')  
time.sleep(1)  
  
print('[Enemy]')  
enemy\_hp = 130  
enemy\_att = 30  
print(f'Heath Point: {enemy\_hp}')  
print(f'Attack: {enemy\_att}')  
print('-------------------------------------------------------------------\n')  
time.sleep(1)  
  
while player\_hp >= 0 and enemy\_hp >= 0:  
 enemy\_hp = enemy\_hp - player\_att  
 player\_hp = player\_hp - enemy\_att  
 print(f'You are attacking enemy, [Enemy\'s] remain health point is: {enemy\_hp}')  
 print(f'Enemy is attacking you, [Player\'s] remain health point is: {player\_hp}')  
 print('-------------------------------------------------------------------\n')  
 time.sleep(1)  
 continue  
  
# checking winner for each rounds  
if player\_hp <= 0 and enemy\_hp <= 0:  
 print('You and Enemy were died both !\n\n')  
 time.sleep(1)  
elif player\_hp <= 0:  
 print('Enemy Win !\n\n')  
 time.sleep(1)  
else:  
 print('You Win !\n\n')  
 time.sleep(1)  
  
print('\t\t\t--------Game Over--------\n\n')

**Running result version 1.0**



**Discussion**

In this version we have to take attributes of players and enemies manually. It is an easy version. So, at first, I take some necessary variables and give attributes manually. After that display their information and the fighting process is started. At the end of the fighting, whose health point is less than 0 first, he will be loose. If both health point is less than zero al the same time then both are died.

**Flowchart for version 2.0**



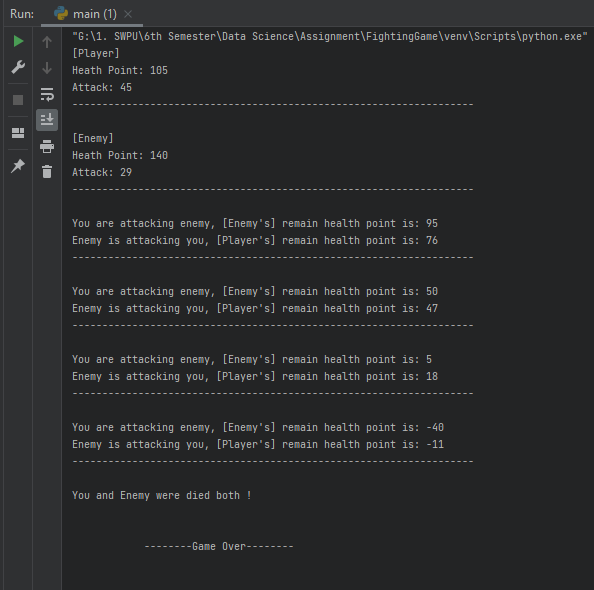
**Source code for version 2.0**

import random  
import time  
  
print('[Player]')  
player\_hp = random.randint(100, 150)  
player\_att = random.randint(20, 50)  
print(f'Heath Point: {player\_hp}')  
print(f'Attack: {player\_att}')  
print('-------------------------------------------------------------------\n')  
time.sleep(1)  
  
print('[Enemy]')  
enemy\_hp = random.randint(100, 150)  
enemy\_att = random.randint(20, 50)  
print(f'Heath Point: {enemy\_hp}')  
print(f'Attack: {enemy\_att}')  
print('-------------------------------------------------------------------\n')  
time.sleep(1)  
  
while player\_hp >= 0 and enemy\_hp >= 0:  
 enemy\_hp = enemy\_hp - player\_att  
 player\_hp = player\_hp - enemy\_att  
 print(f'You are attacking enemy, [Enemy\'s] remain health point is: {enemy\_hp}')  
 print(f'Enemy is attacking you, [Player\'s] remain health point is: {player\_hp}')  
 print('-------------------------------------------------------------------\n')  
 time.sleep(1)  
 continue  
  
# checking winner for each rounds  
if player\_hp <= 0 and enemy\_hp <= 0:  
 print('You and Enemy were died both !\n\n')  
 time.sleep(1)  
elif player\_hp <= 0:  
 print('Enemy Win !\n\n')  
 time.sleep(1)  
else:  
 print('You Win !\n\n')  
 time.sleep(1)  
  
print('\t\t\t--------Game Over--------\n\n')

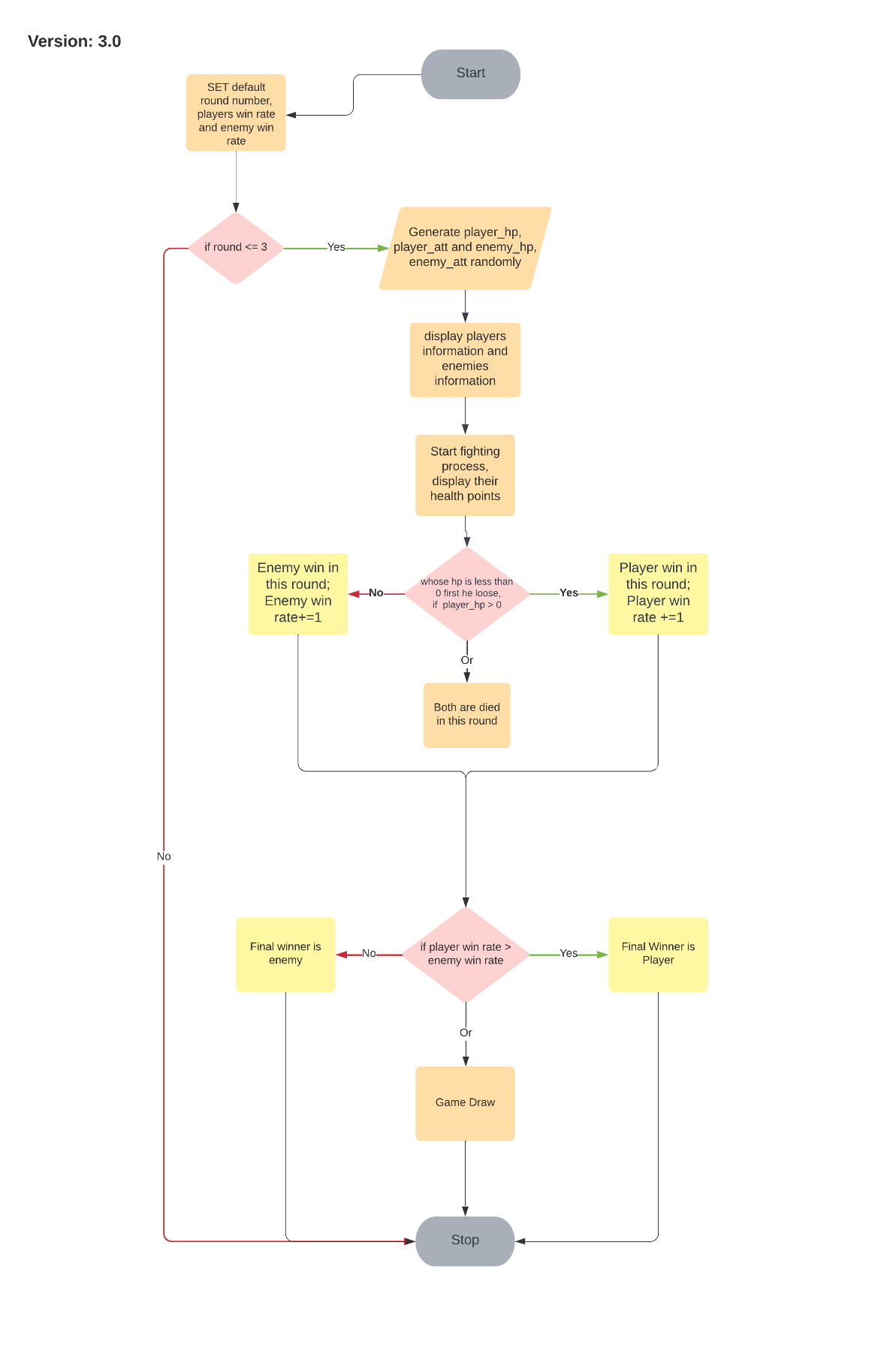
**Discussion**

In this version we have to take attributes of players and enemies randomly. It is also an easy version. So, at first, I generate attributes randomly of players and enemies. After that display their information and the fighting process is started. At the end of the fighting, whose health point is less than 0 first, he will be loose. If both health point is less than zero al the same time then both are died, same as version 1.0

**Running result version 2.0**



**Flowchart for version 3.0:**



**Source code for version 3.0**

import random  
import time  
  
rounds = 1  
enemy\_win = 0  
player\_win = 0  
  
while rounds <= 3:  
 print(f'\n\t\t\t--------Now it\'s Round {rounds} --------\n')  
 time.sleep(1)  
  
 print('[Player]')  
 player\_hp = random.randint(100, 150)  
 player\_att = random.randint(20, 50)  
 print(f'Heath Point: {player\_hp}')  
 print(f'Attack: {player\_att}')  
 print('-------------------------------------------------------------------\n')  
 time.sleep(1)  
  
 print('[Enemy]')  
 enemy\_hp = random.randint(100, 150)  
 enemy\_att = random.randint(20, 50)  
 print(f'Heath Point: {enemy\_hp}')  
 print(f'Attack: {enemy\_att}')  
 print('-------------------------------------------------------------------\n')  
 time.sleep(1)  
  
 while player\_hp >= 0 and enemy\_hp >= 0:  
 enemy\_hp = enemy\_hp - player\_att  
 player\_hp = player\_hp - enemy\_att  
 print(f'You are attacking enemy, [Enemy\'s] remain health point is: {enemy\_hp}')  
 print(f'Enemy is attacking you, [Player\'s] remain health point is: {player\_hp}')  
 print('-------------------------------------------------------------------\n')  
 time.sleep(1)  
 continue  
  
 # checking winner for each rounds  
 if player\_hp <= 0 and enemy\_hp <= 0:  
 print('You and Enemy were died both !\n\n')  
 elif player\_hp <= 0:  
 print('Enemy Win !\n\n')  
 enemy\_win += 1  
 else:  
 print('You Win !\n\n')  
 player\_win += 1  
 time.sleep(1)  
  
 rounds += 1  
  
print('\t\t\t--------Game Over--------\n\n')  
  
# checking final winner and looser  
if player\_win == enemy\_win:  
 print('Game draw !\n')  
elif player\_win > enemy\_win:  
 print('Final Winner: You !\nLooser: Enemy !\n')  
else:  
 print('Final Winner: Enemy !\nLooser: You !\n')

**Discussion**

In this version, we have to play three rounds of this game. So, at first, check the rounds, if rounds are less than 3 then game will continue otherwise game will end. we have to take attributes of players and enemies randomly as like version 2.0. So, generate attributes randomly of players and enemies. After that display their information and the fighting process is started. At the end of the fighting, whose health point is less than 0 first, he will be loose. If both health point is less than zero al the same time then both are died, same as version 1.0 and version 2.0. After finishing all rounds, we have final winner. Final winner is the best of 3 sets. If player wins more than enemy then final winner is player, or final winner is enemy. If player and enemy both winning rate is same, then game is draw.

**Running results**

