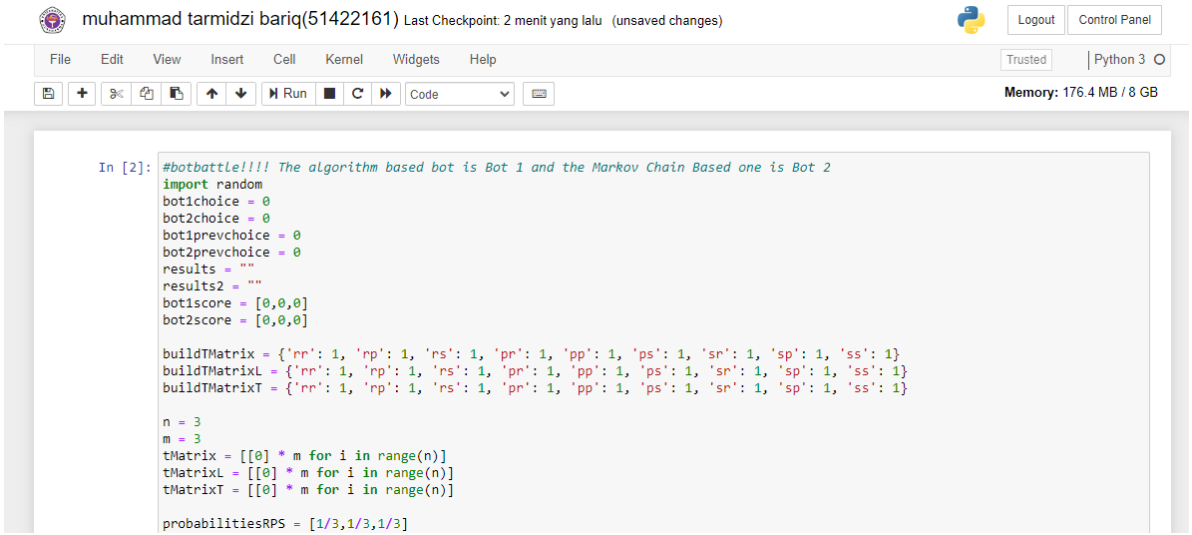
TUGAS PRAKTIKUM PERTEMUAN 5

NAMA : MUHAMMAD TARMIDZI BARIQ
KELAS : 1IA13
NPM : 51422161




```
def init():
    global bot1prevchoice
    global bot1choice
    global bot2prevchoice
    global bot2choice
    global results
    global results2

    bot1prevchoice, bot1choice = bot1(0,0,"Tied!") #inital assuming previous choice was rock and current choice is rock
    bot2prevchoice, bot2choice = bot2(0,0,"Tied!") #inital assuming previous choice was rock and current choice is rock (result
    results = checkWin(bot1choice,bot2choice)
    if (results == "Win!"):
      results2 = "Lose!"
    elif (results == "Lose!"):
      results2 = "Win!"
    else:
      results2 = "Tied!"

    fight(100000)

def fight(rounds):
```

```
def fight(rounds):
    global bot1prevchoice
    global bot1choice
    global bot2prevchoice
    global bot2choice
    global results
    global results2
    global bot1score
    global bot2score
    choose = ["Rock","Paper","Scissors"]
    for i in range(0,rounds):
        bot1prevchoice, bot1choice = bot1(bot2prevchoice,bot2choice,results2)
        #print ("Bot 1 chose %s" % choose[bot1choice])
        bot2prevchoice, bot2choice = bot2(bot1prevchoice,bot1choice,results)
        #print ("Bot 2 chose %s" % choose[bot2choice])
        #print (buildTMatrix)
        #print (buildTMatrixL)
        #print (buildTMatrixT)
        results = checkWin(bot1choice,bot2choice)
```

```
        results = checkWin(bot1choice,bot2choice)
        if (results == "Win!"):
          results2 = "Lose!"
        elif (results == "Lose!"):
          results2 = "Win!"
        else:
          results2 = "Tied!"
        #print (results)
        if (results == "Win!"):
            bot2score[1] += 1
            bot1score[0] += 1
        elif (results == "Lose!"):
            bot1score[1] += 1
            bot2score[0] += 1
        else:
            bot2score[2] += 1
            bot1score[2] += 1
    print ("Bot1 won %s times, lost %s times, and tied %s times.\n\nTo check:\n\nBot2 won %s times, lost %s times, and tied %s ti

def bot1(prev,choit,res):
    choices = ["Rock", "Paper", "Scissors"]
    prevChoice = prev
    choice = choit
    prevMachineChoice = ""
    result = res
    streak = 0
    won = 0
    alt = 0
    numoff = 0

    if (prevChoice == choice):
        streak += 1
    else:
        streak -= 1
        if (streak < 0):
            streak = 0
    winprev = prevChoice + 1
    if (winprev > 2):
        winprev = 0
    if (choice == winprev):
        alt += 1
    else:
        alt -= 1
        if (alt < 0):
            alt = 0
    if (alt > 3):
        machineChoice = winprev + 1
        if (machineChoice > 2):
            machineChoice = 0
    elif (streak > 3):
        machineChoice = prevChoice - 2
        if (machineChoice < 0):
            machineChoice += 3
    elif (won > 9):
        machineChoice = random.randint(0, 2)
    elif (won > 3 and won < 10):
        machineChoice = prevChoice
    else:
        if (result == "Win!"):
            machineChoice = prevChoice - 2
            if (machineChoice < 0):
                machineChoice += 3
        elif (result == "Lose!"):
            machineChoice = prevChoice + 1
            if (machineChoice > 2):
                machineChoice -= 3
            machineChoice -= 2
            if (machineChoice < 0):
                machineChoice += 3
        else:
            machineChoice = random.randint(0, 2)

    result = checkWin(choice, machineChoice)

    if (result == "Win!"):
        won += 1
    else:
        won -= 2
        if (won < 0):
            won = 0
    return choit, machineChoice
```

```
def checkWin(user, machine):
    win = False
    tie = False
    if (user == 0):
        if (machine == 2):
            win = True
            tie = False
        elif (machine == 1):
            win = False
            tie = False
        elif (user == 0):
            tie = True
        else:
            print ("Something wierd happened and machine was: %s" % machine)
    elif (user == 1):
        if (machine == 0):
            win = True
            tie = False
        elif (machine == 2):
            win = False
            tie = False
        elif (machine == 1):
            tie = True
        else:
            print ("Something wierd happened and machine was: %s" % machine)
    else:
        if (machine == 1):
            win = True
            tie = False
        elif (machine == 0):
            win = False
            tie = False
        elif (machine == 2):
            tie = True
        else:
            print ("Something wierd happened and machine was: %s" % machine)

    if (tie == True):
        return "Tied!"
    elif (win):
        return "Win!"
    else:
        return "Lose!"

def bot2(previ,choit,res):
    global probabilitiesRPS
    choices = ["Rock","Paper","Scissors"]
    choi = ['r','p','s']
    prevChoice = previ
    probRock = 0
    probPaper = 0
    probScissors = 0
    result = res

    choice = choit
    transMatrix = buildTransitionProbabilities(prevChoice,choice,result)
    machineChoice = random.randint(1, 100)
    probabilitiesRPS[0] = transMatrix[prevChoice][0]
    probabilitiesRPS[1] = transMatrix[prevChoice][1]
    probabilitiesRPS[2] = transMatrix[prevChoice][2]
    rangeR = probabilitiesRPS[0] * 100
    rangeP = probabilitiesRPS[1] * 100 + rangeR
    if (machineChoice <= rangeR):
        machineChoice = 1
    elif (machineChoice <= rangeP):
        machineChoice = 2
    else:
        machineChoice = 0
    return choit, machineChoice
```

```
def buildTransitionProbabilities(pC,c,winloss):
    global buildTMatrix
    global buildTMatrixL
    global buildTMatrixT
    choi = ['r','p','s']

    if winloss == "Win!":
        for i, x in buildTMatrix.items():
          if ('%s%s' % (choi[pC],choi[c]) == i):
            buildTMatrix['%s%s' % (choi[pC], choi[c])] += 1
    elif winloss == "Tied!":
        for i, x in buildTMatrixT.items():
          if ('%s%s' % (choi[pC],choi[c]) == i):
            buildTMatrixT['%s%s' % (choi[pC], choi[c])] += 1
    else:
        for i, x in buildTMatrixL.items():
          if ('%s%s' % (choi[pC],choi[c]) == i):
            buildTMatrixL['%s%s' % (choi[pC], choi[c])] += 1

    return buildTransitionMatrix(winloss)

def buildTransitionMatrix(winlosstwo):
    global tMatrix
    global tMatrixL
    global tMatrixT

    if winlosstwo == "Win!":
      rock = buildTMatrix['rr'] + buildTMatrix['rs'] +buildTMatrix['rp']
      paper = buildTMatrix['pr'] + buildTMatrix['ps'] +buildTMatrix['pp']
      scissors = buildTMatrix['sr'] + buildTMatrix['ss'] +buildTMatrix['sp']
      choi = ['r','p','s']
      for row_index, row in enumerate(tMatrix):
        for col_index, item in enumerate(row):
          a = int(buildTMatrix['%s%s' % (choi[row_index],choi[col_index])])
          if (row_index == 0):
            c = a/rock
          elif (row_index == 1):
            c = a/paper
          else:
            c = a/scissors
          row[col_index] = float(c)
      return (tMatrix)
    elif winlosstwo == "Tied!":
      rock = buildTMatrixT['rr'] + buildTMatrixT['rs'] +buildTMatrixT['rp']
      paper = buildTMatrixT['pr'] + buildTMatrixT['ps'] +buildTMatrixT['pp']
      scissors = buildTMatrixT['sr'] + buildTMatrixT['ss'] +buildTMatrixT['sp']
      choi = ['r','p','s']
      for row_index, row in enumerate(tMatrixT):
        for col_index, item in enumerate(row):
          a = int(buildTMatrixT['%s%s' % (choi[row_index],choi[col_index])])
          if (row_index == 0):
            c = a/rock
          elif (row_index == 1):
            c = a/paper
          else:
            c = a/scissors
          row[col_index] = float(c)
      return (tMatrixT)

    else:
      rock = buildTMatrixL['rr'] + buildTMatrixL['rs'] +buildTMatrixL['rp']
      paper = buildTMatrixL['pr'] + buildTMatrixL['ps'] +buildTMatrixL['pp']
      scissors = buildTMatrixL['sr'] + buildTMatrixL['ss'] +buildTMatrixL['sp']
      choi = ['r','p','s']
      for row_index, row in enumerate(tMatrixL):
        for col_index, item in enumerate(row):
          a = int(buildTMatrixL['%s%s' % (choi[row_index],choi[col_index])])
          if (row_index == 0):
            c = a/rock
          elif (row_index == 1):
            c = a/paper
          else:
            c = a/scissors
          row[col_index] = float(c)
      return (tMatrixL)

init()
```

```
Bot1 won 510 times, lost 98980 times, and tied 510 times.

To check:

Bot2 won 98980 times, lost 510 times, and tied 510 times.
```