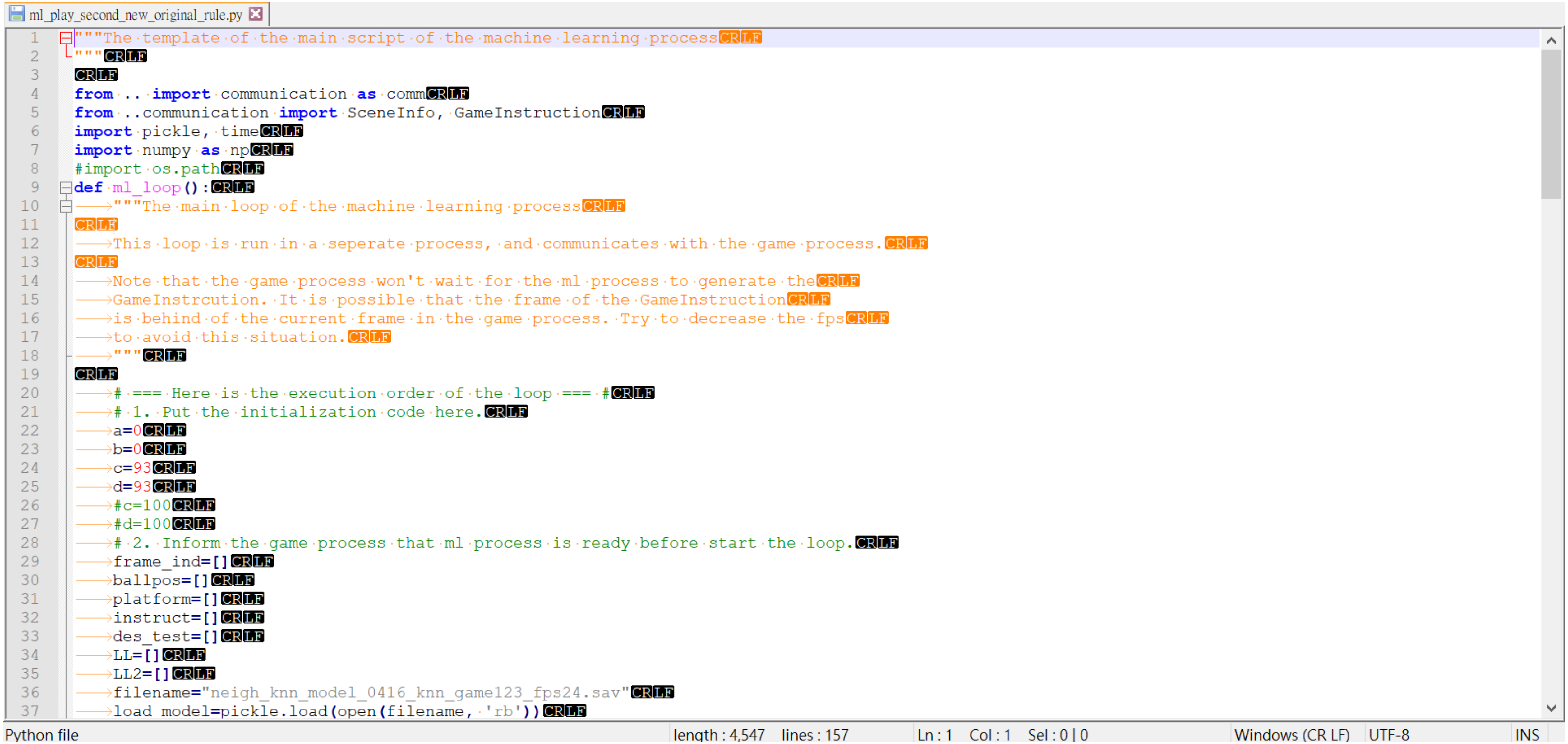
05/23 機器學習 打磚塊 正式比賽

張君豪

New Original Rule (ver.2)

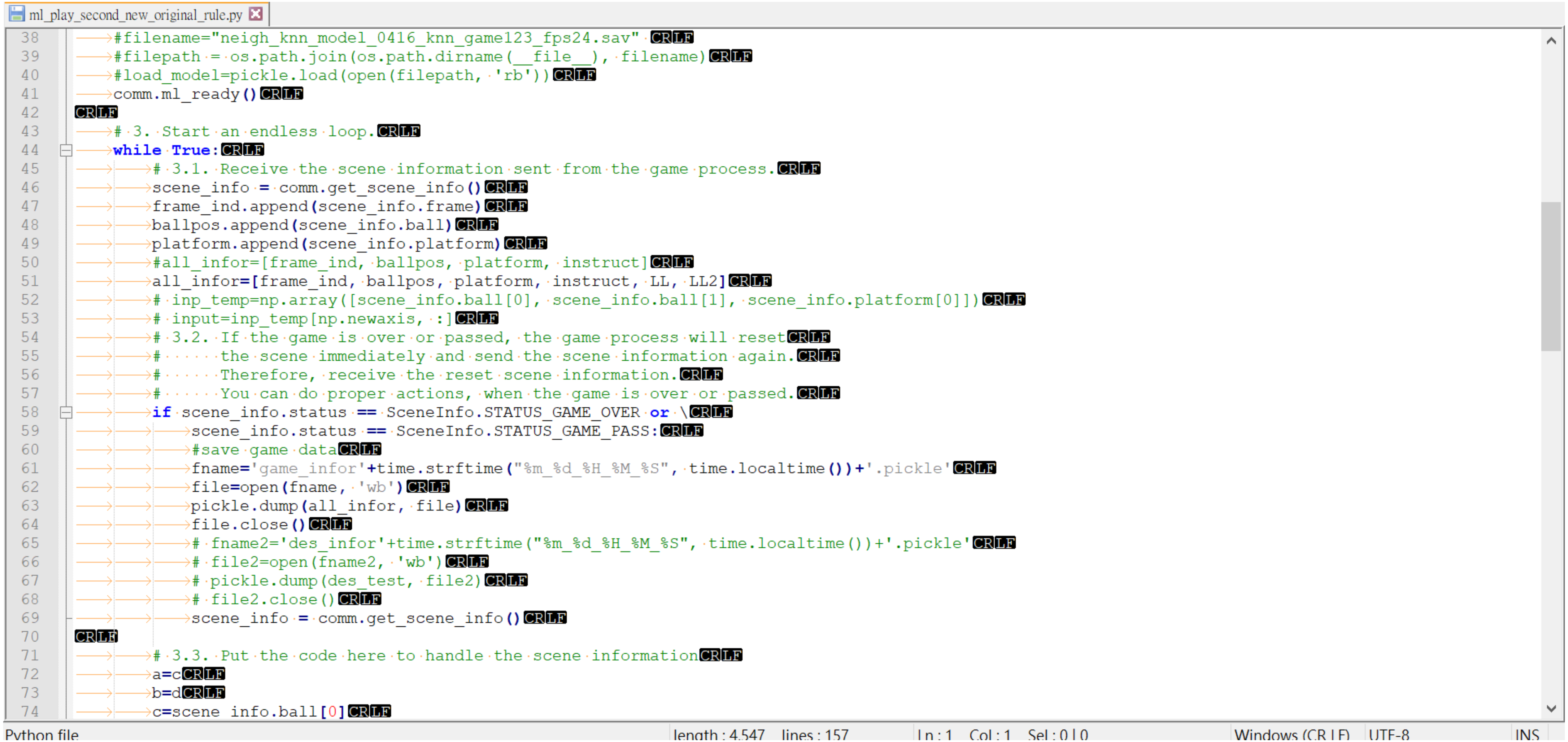
- 公式算出球的落點
- 再讓板子移到接的到球的地方



ml_play_second_new_original_rule.py

```python
"""The template of the main script of the machine learning process
"""

from .. import communication as comm
from ..communication import SceneInfo, GameInstruction
import pickle, time
import numpy as np
#import os.path
def ml_loop():
	"""The main loop of the machine learning process

	This loop is run in a seperate process, and communicates with the game process.

	Note that the game process won't wait for the ml process to generate the
	GameInstrcution. It is possible that the frame of the GameInstruction
	is behind of the current frame in the game process. Try to decrease the fps
	to avoid this situation.
	"""

	# === Here is the execution order of the loop === #
	# 1. Put the initialization code here.
	a=0
	b=0
	c=93
	d=93
	#c=100
	#d=100
	# 2. Inform the game process that ml process is ready before start the loop.
	frame_ind=[]
	ballpos=[]
	platform=[]
	instruct=[]
	des_test=[]
	LL=[]
	LL2=[]
	filename="neigh_knn_model_0416_knn_game123_fps24.sav"
	load_model=pickle.load(open(filename, 'rb'))
```

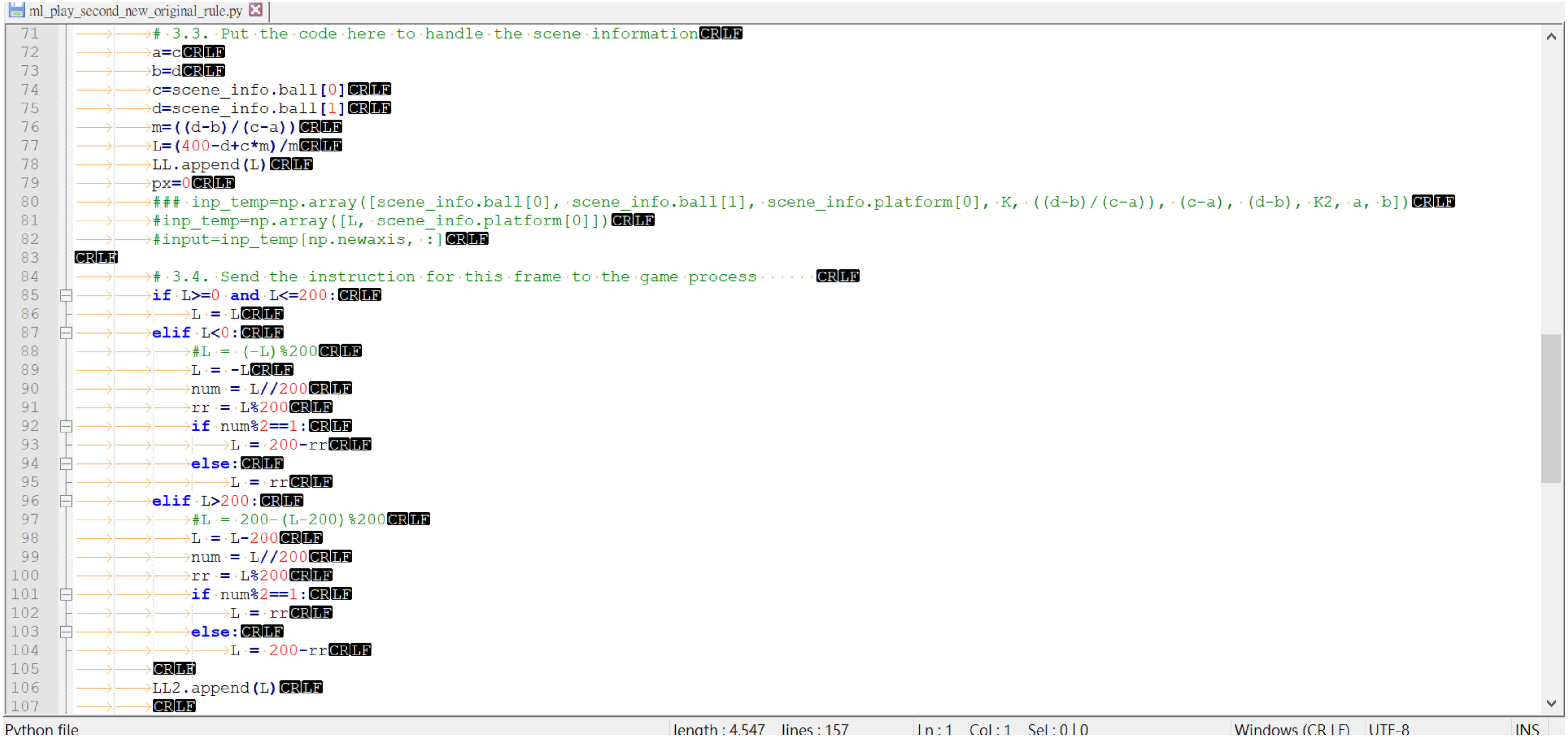
Python file   length : 4,547   lines : 157   Ln : 1   Col : 1   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS



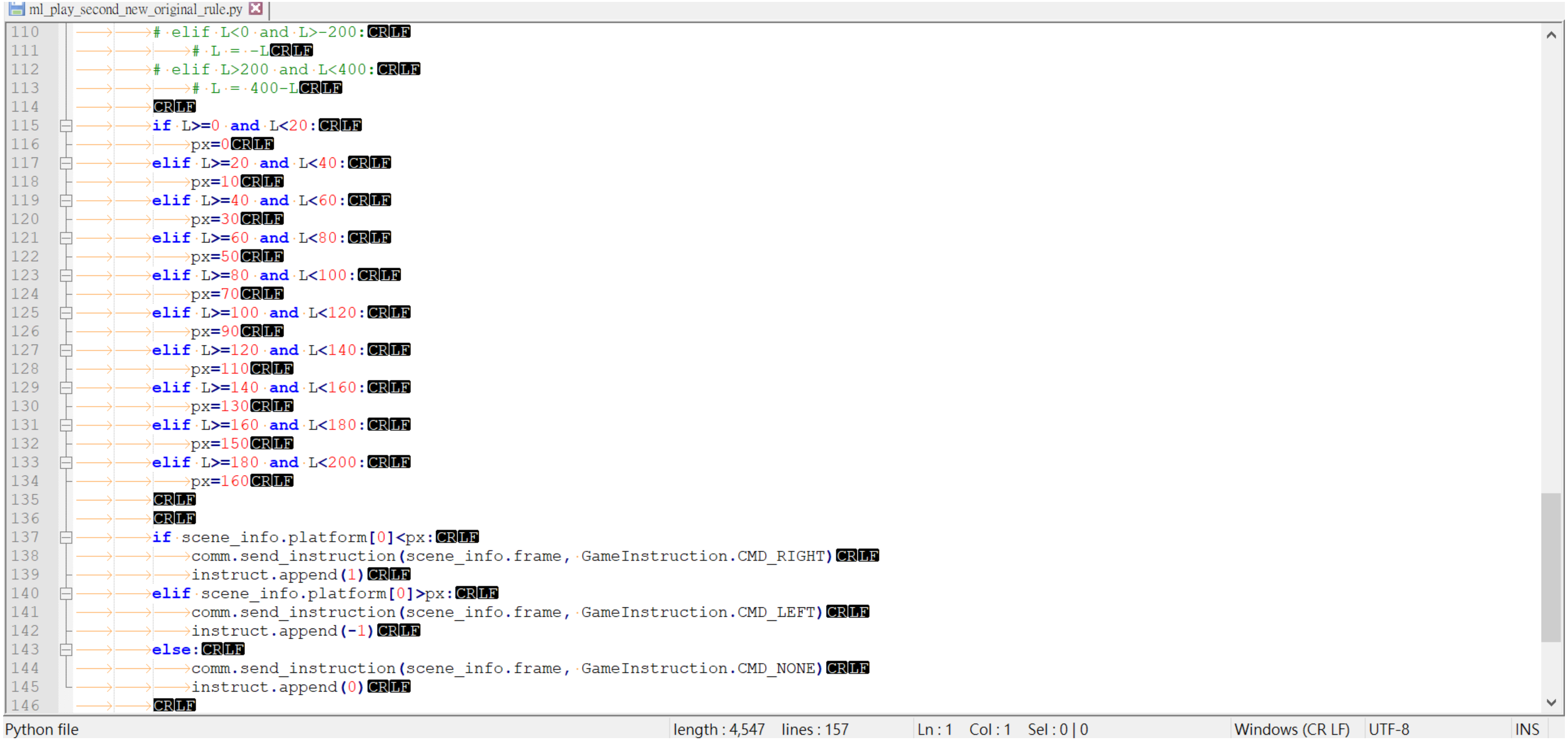
ml_play_second_new_original_rule.py

```python
    #filename="neigh_knn_model_0416_knn_game123_fps24.sav"
    #filepath = os.path.join(os.path.dirname(__file__), filename)
    #load_model=pickle.load(open(filepath, 'rb'))
    comm.ml_ready()

    # 3. Start an endless loop.
    while True:
        # 3.1. Receive the scene information sent from the game process.
        scene_info = comm.get_scene_info()
        frame_ind.append(scene_info.frame)
        ballpos.append(scene_info.ball)
        platform.append(scene_info.platform)
        #all_infor=[frame_ind, ballpos, platform, instruct]
        all_infor=[frame_ind, ballpos, platform, instruct, LL, LL2]
        # inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0]])
        # input=inp_temp[np.newaxis, :]
        # 3.2. If the game is over or passed, the game process will reset
        #      the scene immediately and send the scene information again.
        #      Therefore, receive the reset scene information.
        #      You can do proper actions, when the game is over or passed.
        if scene_info.status == SceneInfo.STATUS_GAME_OVER or \
            scene_info.status == SceneInfo.STATUS_GAME_PASS:
            #save game data
            fname='game_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle'
            file=open(fname, 'wb')
            pickle.dump(all_infor, file)
            file.close()
            # fname2='des_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle'
            # file2=open(fname2, 'wb')
            # pickle.dump(des_test, file2)
            # file2.close()
            scene_info = comm.get_scene_info()

        # 3.3. Put the code here to handle the scene information
        a=c
        b=d
        c=scene_info.ball[0]
```

Python file   length : 4,547   lines : 157   Ln : 1   Col : 1   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS


ml_play_second_new_original_rule.py
Python file
length : 4,547   lines : 157
Ln : 1   Col : 1   Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

```
        # 3.3. Put the code here to handle the scene information
        a=c
        b=d
        c=scene_info.ball[0]
        d=scene_info.ball[1]
        m=((d-b)/(c-a))
        L=(400-d+c*m)/m
        LL.append(L)
        px=0
        ### inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0], K, ((d-b)/(c-a)), (c-a), (d-b), K2, a, b])
        #inp_temp=np.array([L, scene_info.platform[0]])
        #input=inp_temp[np.newaxis, :]

        # 3.4. Send the instruction for this frame to the game process
        if L>=0 and L<=200:
            L = L
        elif L<0:
            #L = (-L)%200
            L = -L
            num = L//200
            rr = L%200
            if num%2==1:
                L = 200-rr
            else:
                L = rr
        elif L>200:
            #L = 200-(L-200)%200
            L = L-200
            num = L//200
            rr = L%200
            if num%2==1:
                L = rr
            else:
                L = 200-rr

        LL2.append(L)

```

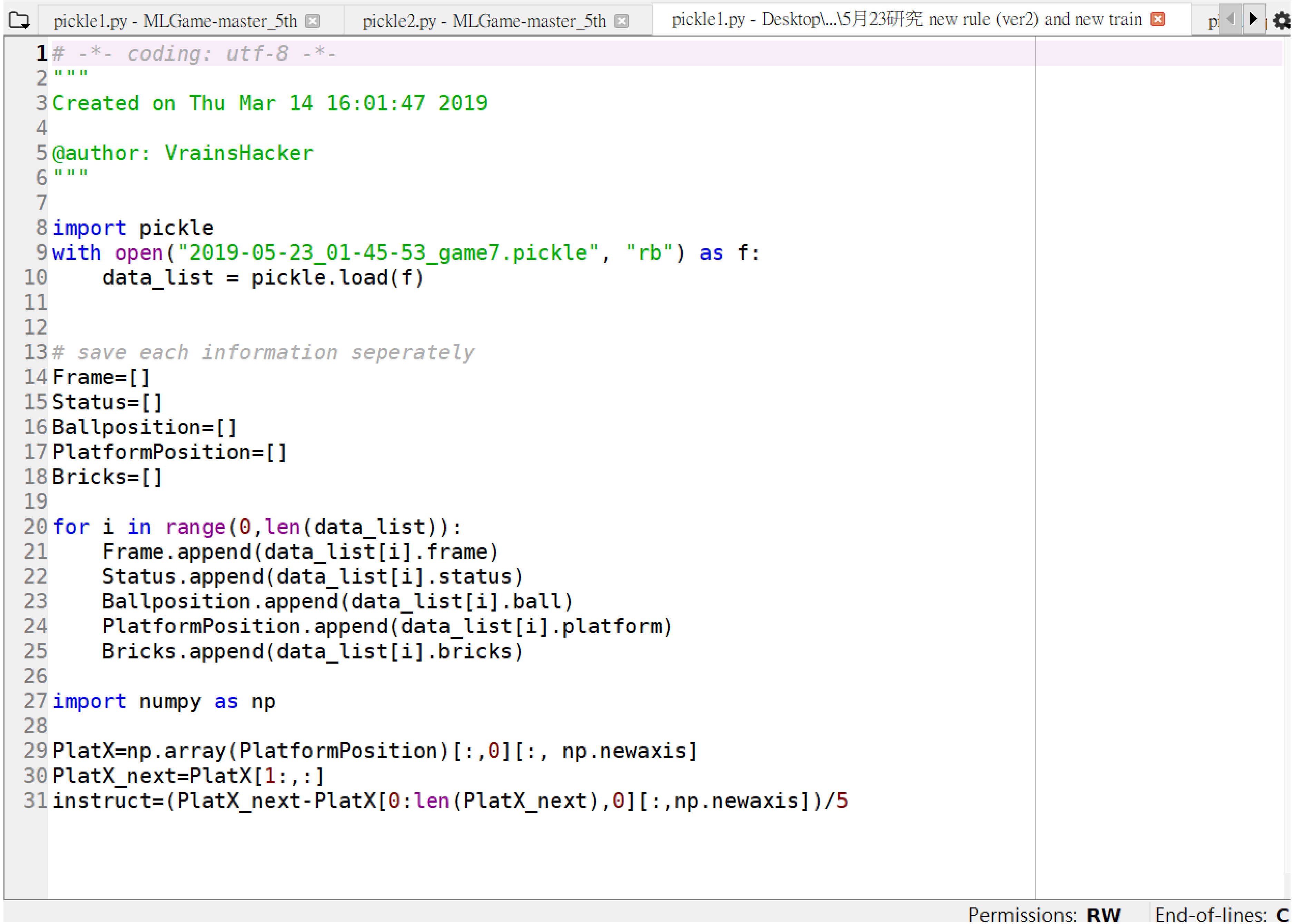

ml_play_second_new_original_rule.py
Python file
length : 4,547   lines : 157
Ln : 1   Col : 1   Sel : 0 | 0
Windows (CR LF)
UTF-8
INS

```
        # elif L<0 and L>-200:
            # L = -L
        # elif L>200 and L<400:
            # L = 400-L

        if L>=0 and L<20:
            px=0
        elif L>=20 and L<40:
            px=10
        elif L>=40 and L<60:
            px=30
        elif L>=60 and L<80:
            px=50
        elif L>=80 and L<100:
            px=70
        elif L>=100 and L<120:
            px=90
        elif L>=120 and L<140:
            px=110
        elif L>=140 and L<160:
            px=130
        elif L>=160 and L<180:
            px=150
        elif L>=180 and L<200:
            px=160


        if scene_info.platform[0]<px:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_RIGHT)
            instruct.append(1)
        elif scene_info.platform[0]>px:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_LEFT)
            instruct.append(-1)
        else:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_NONE)
            instruct.append(0)

```

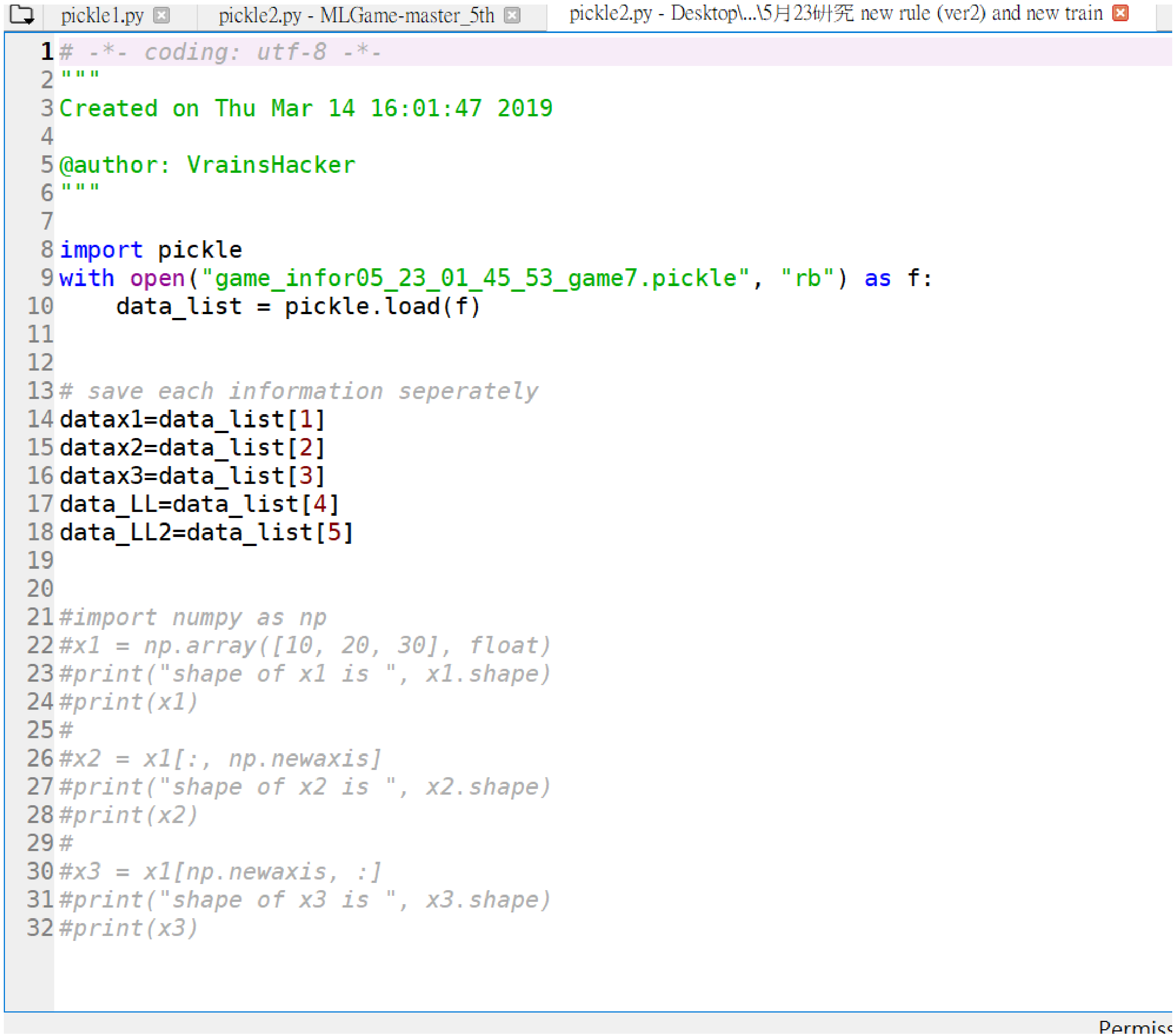
用pickle取data放進.csv檔

- Feature X:
- 球的落點、板子的x座標

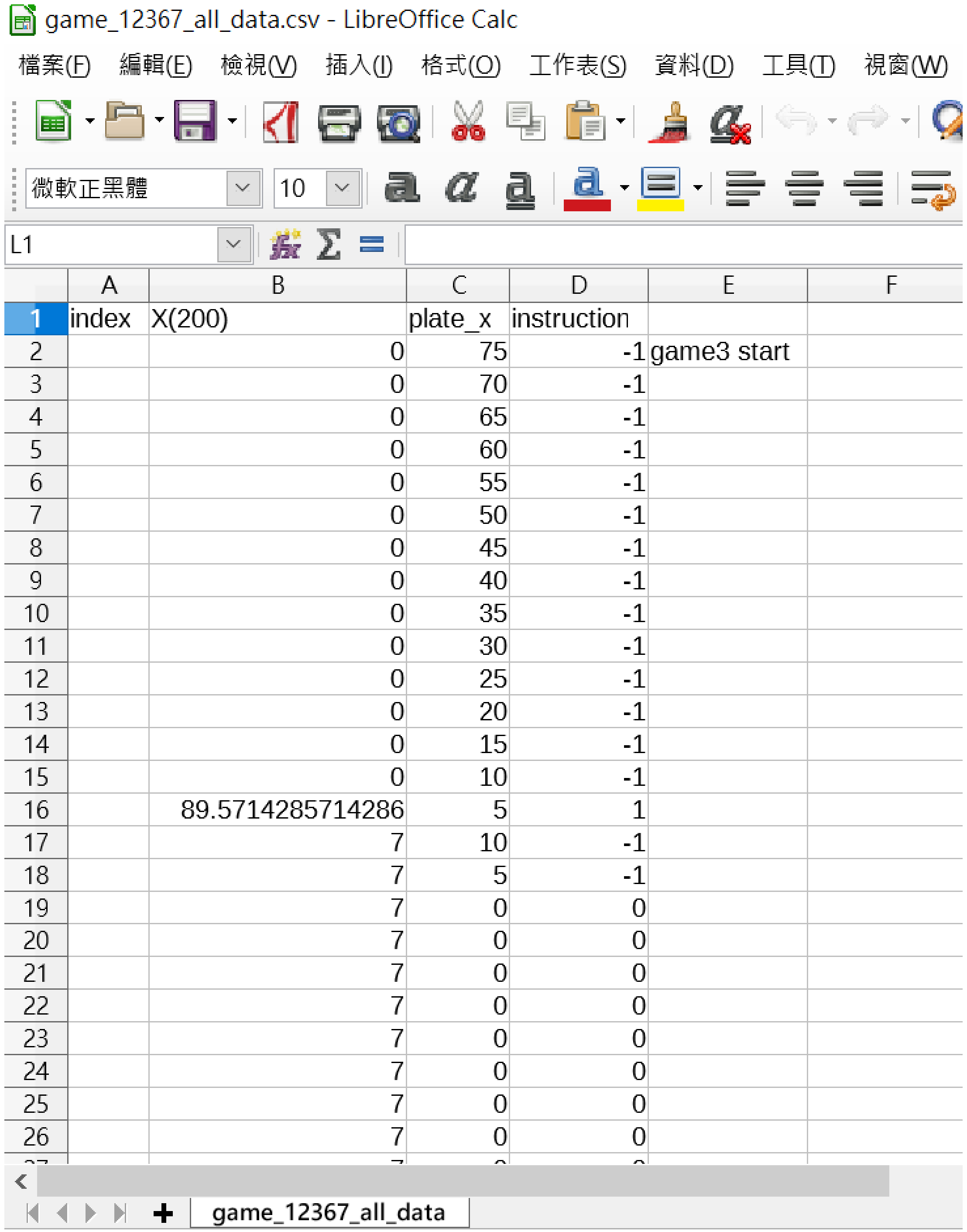
- Predict Y:
- 移動板子的指令 (0、1、 –1)


pickle1.py - MLGame-master_5th
pickle2.py - MLGame-master_5th
pickle1.py - Desktop\...\5月23研究 new rule (ver2) and new train
pi
Permissions: RW
End-of-lines: C

```python
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 16:01:47 2019

@author: VrainsHacker
"""

import pickle
with open("2019-05-23_01-45-53_game7.pickle", "rb") as f:
    data_list = pickle.load(f)


# save each information seperately
Frame=[]
Status=[]
Ballposition=[]
PlatformPosition=[]
Bricks=[]

for i in range(0,len(data_list)):
    Frame.append(data_list[i].frame)
    Status.append(data_list[i].status)
    Ballposition.append(data_list[i].ball)
    PlatformPosition.append(data_list[i].platform)
    Bricks.append(data_list[i].bricks)

import numpy as np

PlatX=np.array(PlatformPosition)[:,0][:, np.newaxis]
PlatX_next=PlatX[1:,:]
instruct=(PlatX_next-PlatX[0:len(PlatX_next),0][:,np.newaxis])/5
```


pickle1.py
pickle2.py - MLGame-master_5th
pickle2.py - Desktop\...\5月23研究 new rule (ver2) and new train
Permiss

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 14 16:01:47 2019

@author: VrainsHacker
"""

import pickle
with open("game_infor05_23_01_45_53_game7.pickle", "rb") as f:
    data_list = pickle.load(f)


# save each information seperately
datax1=data_list[1]
datax2=data_list[2]
datax3=data_list[3]
data_LL=data_list[4]
data_LL2=data_list[5]


#import numpy as np
#x1 = np.array([10, 20, 30], float)
#print("shape of x1 is ", x1.shape)
#print(x1)
#
#x2 = x1[:, np.newaxis]
#print("shape of x2 is ", x2.shape)
#print(x2)
#
#x3 = x1[np.newaxis, :]
#print("shape of x3 is ", x3.shape)
#print(x3)
```



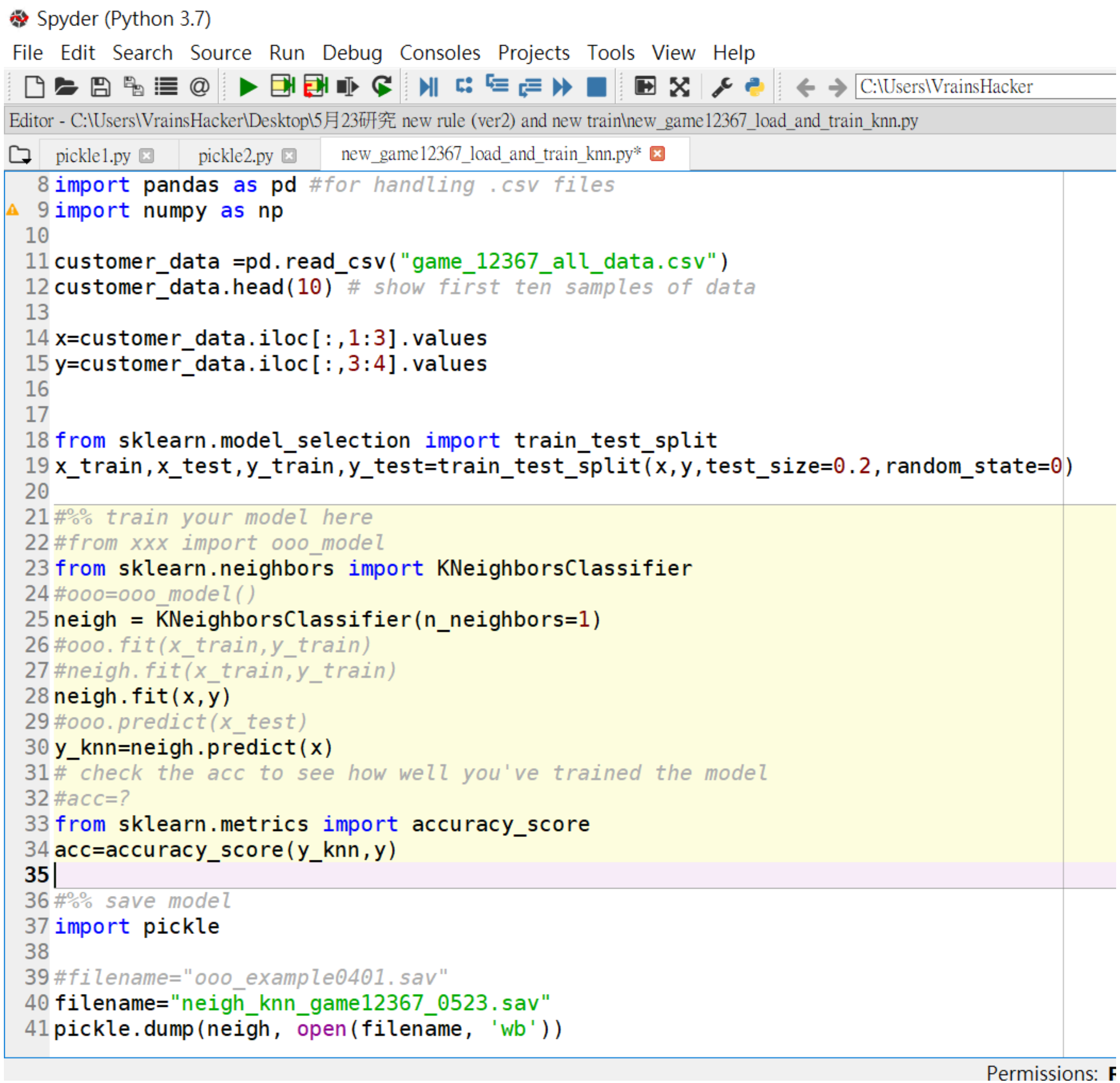
	A	B	C	D	E	F
1	index	X(200)	plate_x	instruction		
2		0	75	-1	game3 start	
3		0	70	-1		
4		0	65	-1		
5		0	60	-1		
6		0	55	-1		
7		0	50	-1		
8		0	45	-1		
9		0	40	-1		
10		0	35	-1		
11		0	30	-1		
12		0	25	-1		
13		0	20	-1		
14		0	15	-1		
15		0	10	-1		
16		89.5714285714286	5	1		
17		7	10	-1		
18		7	5	-1		
19		7	0	0		
20		7	0	0		
21		7	0	0		
22		7	0	0		
23		7	0	0		
24		7	0	0		
25		7	0	0		
26		7	0	0		

Load data and train model

- 用knn

- Feature X:
- 球的落點、板子的x座標

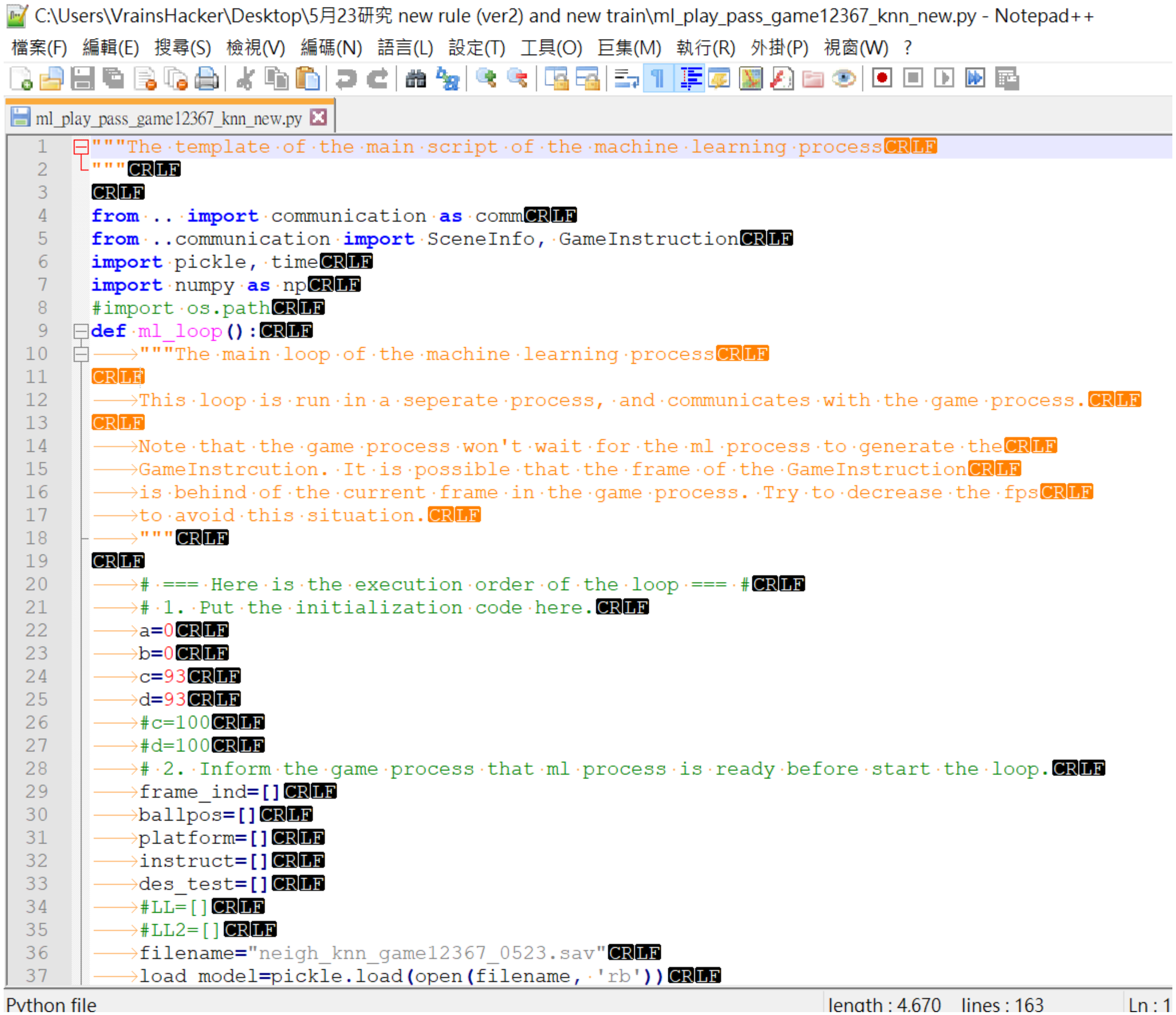
- Predict Y:
- 移動板子的指令 (0、1、 –1)



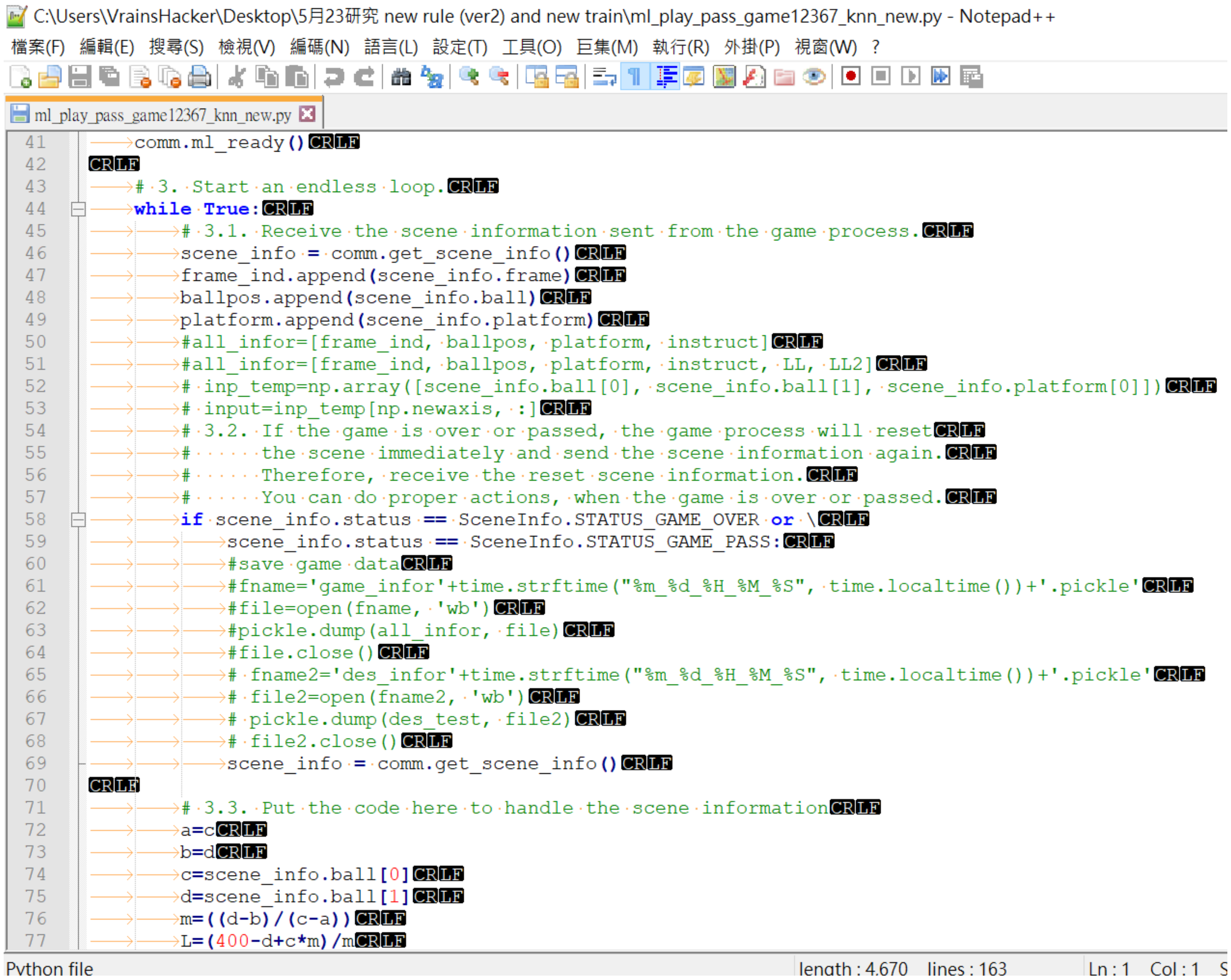
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\VrainsHacker
Editor - C:\Users\VrainsHacker\Desktop\5月23研究 new rule (ver2) and new train\new_game12367_load_and_train_knn.py
pickle1.py
pickle2.py
new_game12367_load_and_train_knn.py*
Permissions: R

```python
import pandas as pd #for handling .csv files
import numpy as np

customer_data =pd.read_csv("game_12367_all_data.csv")
customer_data.head(10) # show first ten samples of data

x=customer_data.iloc[:,1:3].values
y=customer_data.iloc[:,3:4].values


from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

#%% train your model here
#from xxx import ooo_model
from sklearn.neighbors import KNeighborsClassifier
#ooo=ooo_model()
neigh = KNeighborsClassifier(n_neighbors=1)
#ooo.fit(x_train,y_train)
#neigh.fit(x_train,y_train)
neigh.fit(x,y)
#ooo.predict(x_test)
y_knn=neigh.predict(x)
# check the acc to see how well you've trained the model
#acc=?
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_knn,y)

#%% save model
import pickle

#filename="ooo_example0401.sav"
filename="neigh_knn_game12367_0523.sav"
pickle.dump(neigh, open(filename, 'wb'))
```

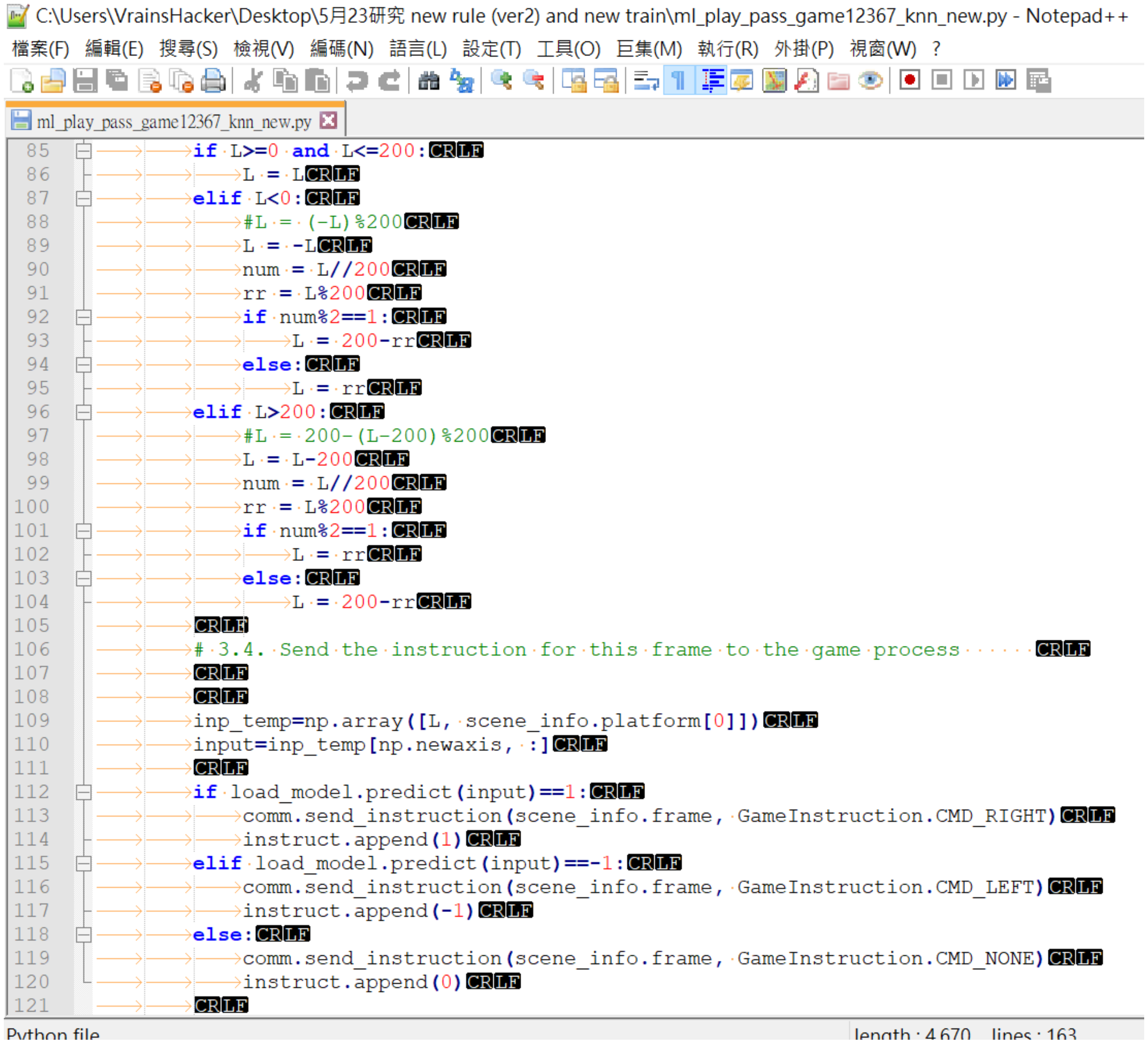
ml_play with machine learning model


C:\Users\VrainsHacker\Desktop\5月23研究 new rule (ver2) and new train\ml_play_pass_game12367_knn_new.py - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
ml_play_pass_game12367_knn_new.py
Python file
length : 4,670 lines : 163
Ln : 1

```python
"""The template of the main script of the machine learning process
"""

from .. import communication as comm
from ..communication import SceneInfo, GameInstruction
import pickle, time
import numpy as np
#import os.path
def ml_loop():
	"""The main loop of the machine learning process

	This loop is run in a seperate process, and communicates with the game process.

	Note that the game process won't wait for the ml process to generate the
	GameInstrcution. It is possible that the frame of the GameInstruction
	is behind of the current frame in the game process. Try to decrease the fps
	to avoid this situation.
	"""

	# === Here is the execution order of the loop === #
	# 1. Put the initialization code here.
	a=0
	b=0
	c=93
	d=93
	#c=100
	#d=100
	# 2. Inform the game process that ml process is ready before start the loop.
	frame_ind=[]
	ballpos=[]
	platform=[]
	instruct=[]
	des_test=[]
	#LL=[]
	#LL2=[]
	filename="neigh_knn_game12367_0523.sav"
	load_model=pickle.load(open(filename, 'rb'))
```


C:\Users\VrainsHacker\Desktop\5月23研究 new rule (ver2) and new train\ml_play_pass_game12367_knn_new.py - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
ml_play_pass_game12367_knn_new.py
Python file
length : 4,670 lines : 163
Ln : 1 Col : 1

```
	comm.ml_ready()

	# 3. Start an endless loop.
	while True:
		# 3.1. Receive the scene information sent from the game process.
		scene_info = comm.get_scene_info()
		frame_ind.append(scene_info.frame)
		ballpos.append(scene_info.ball)
		platform.append(scene_info.platform)
		#all_infor=[frame_ind, ballpos, platform, instruct]
		#all_infor=[frame_ind, ballpos, platform, instruct, LL, LL2]
		# inp_temp=np.array([scene_info.ball[0], scene_info.ball[1], scene_info.platform[0]])
		# input=inp_temp[np.newaxis, :]
		# 3.2. If the game is over or passed, the game process will reset
		#      the scene immediately and send the scene information again.
		#      Therefore, receive the reset scene information.
		#      You can do proper actions, when the game is over or passed.
		if scene_info.status == SceneInfo.STATUS_GAME_OVER or \
			scene_info.status == SceneInfo.STATUS_GAME_PASS:
			#save game data
			#fname='game_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle'
			#file=open(fname, 'wb')
			#pickle.dump(all_infor, file)
			#file.close()
			# fname2='des_infor'+time.strftime("%m_%d_%H_%M_%S", time.localtime())+'.pickle'
			# file2=open(fname2, 'wb')
			# pickle.dump(des_test, file2)
			# file2.close()
			scene_info = comm.get_scene_info()

		# 3.3. Put the code here to handle the scene information
		a=c
		b=d
		c=scene_info.ball[0]
		d=scene_info.ball[1]
		m=((d-b)/(c-a))
		L=(400-d+c*m)/m
```


C:\Users\VrainsHacker\Desktop\5月23研究 new rule (ver2) and new train\ml_play_pass_game12367_knn_new.py - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
ml_play_pass_game12367_knn_new.py

```
        if L>=0 and L<=200:
            L = L
        elif L<0:
            #L = (-L)%200
            L = -L
            num = L//200
            rr = L%200
            if num%2==1:
                L = 200-rr
            else:
                L = rr
        elif L>200:
            #L = 200-(L-200)%200
            L = L-200
            num = L//200
            rr = L%200
            if num%2==1:
                L = rr
            else:
                L = 200-rr

        # 3.4. Send the instruction for this frame to the game process     


        inp_temp=np.array([L, scene_info.platform[0]])
        input=inp_temp[np.newaxis, :]

        if load_model.predict(input)==1:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_RIGHT)
            instruct.append(1)
        elif load_model.predict(input)==-1:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_LEFT)
            instruct.append(-1)
        else:
            comm.send_instruction(scene_info.frame, GameInstruction.CMD_NONE)
            instruct.append(0)

```

result

- Pass game 1 2 3 6 7