

乒乓球 專案管理

指導老師：陳朝烈

成員：電子工程系4甲 0652054

成員：電子工程系4甲 0652074

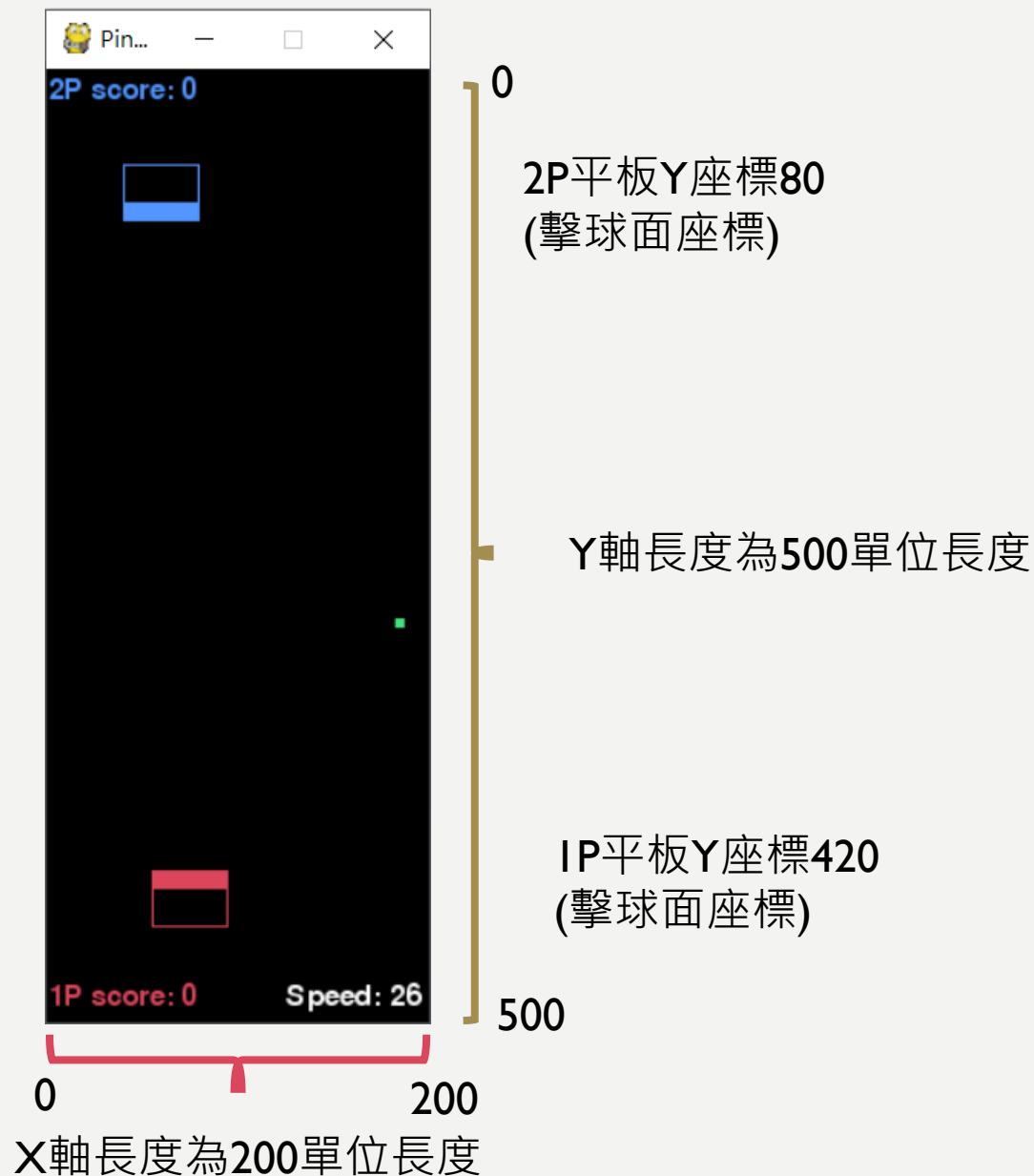
沈易賢

周登豐

專案功能需求-限制需求

- 作業系統: win10版本
- 軟體版本:python3.8
- Mlgame版本為最新8.0版本

專案限制-基本遊戲



- 遊戲規則:
發球為1P往2P移動，球若撞擊到邊界或平板則會反彈，若球移動到1P平板的後方，則判別2P得分，反之亦然。
- 物件大小:
球5x5單位面積
平板40x30單位面積
場地200x500單位面積
- 變動係數:
球的初始速度為每frame移動 ± 7 單位，每過100frames則增加 ± 1 ，如果球速超過40，則此回合為平局遊戲。

專案功能需求-效能需求

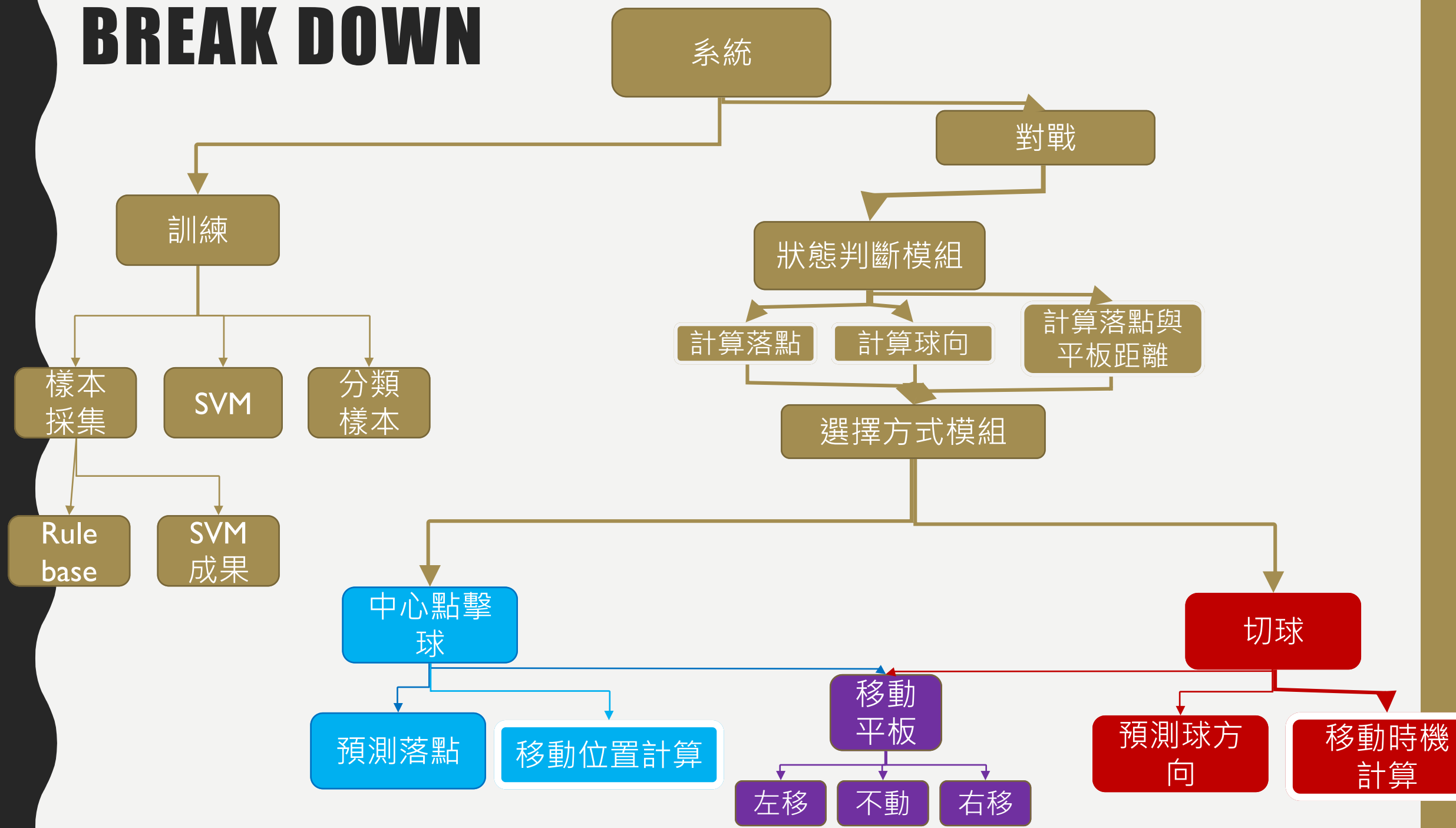
- FPS > 60

需求－功能需求

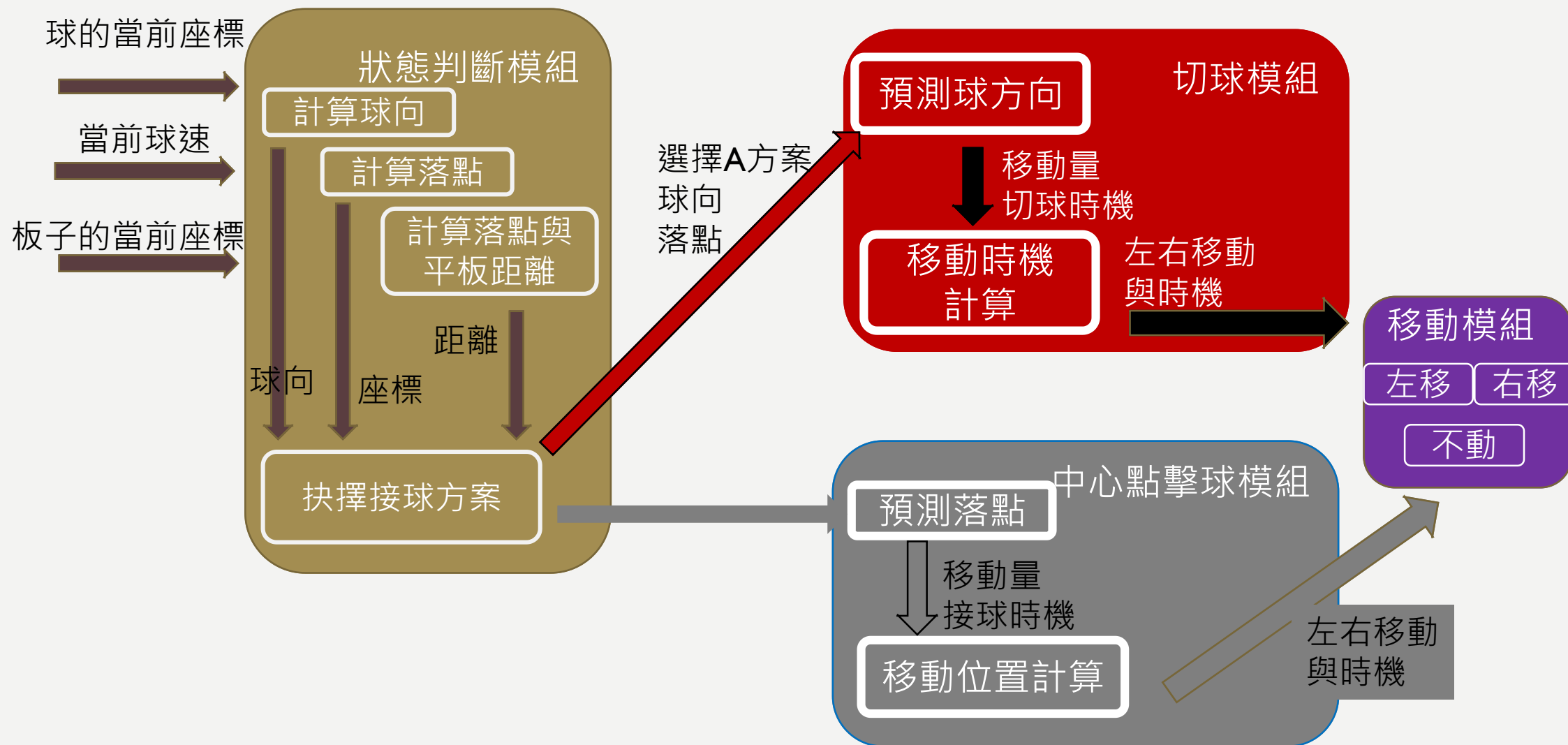
功能需求:

- 預測對方擊球後落點
- 平板以中心接球
- 在速度到達15以前不失誤
- 切球(左右移動直到擊完球)

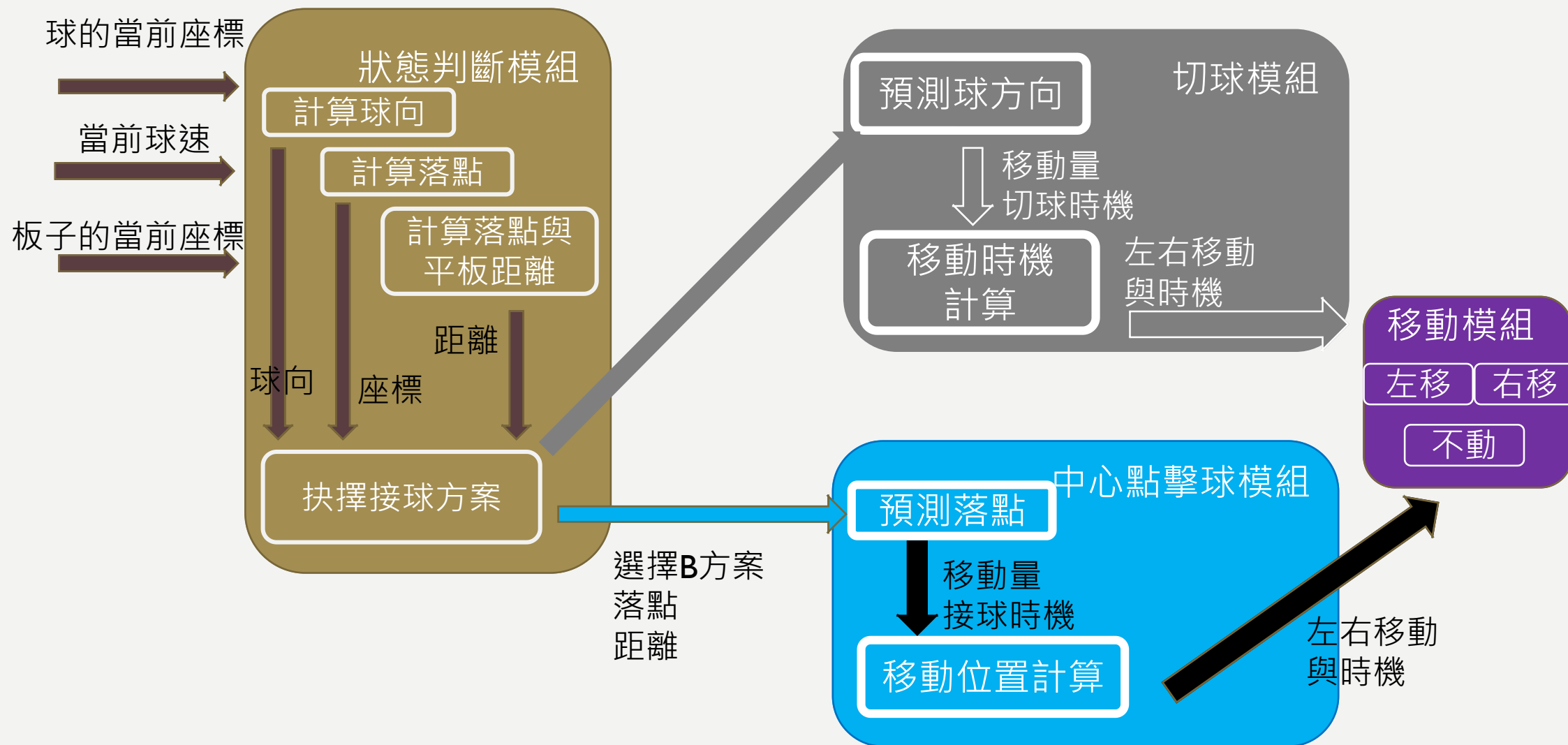
BREAK DOWN



架構圖



架構圖



特徵選取

- 樣本初始化 球座標(X,Y) 板座標(X) 向量(X,Y)
- 可視化
- 聚類 clustering
- Feature 定義

程式

```
import pickle
from sklearn.linear_model import LinearRegression
import numpy as np
import math
import random

filename = "C:\\Users\\User\\Student\\machine-learning-main54\\machine-learning-main\\final\\SVM3.sav"
model = pickle.load(open(filename, 'rb'))
ball_position_history = []
BallPosition=[]
ball_served_random=random.randrange(1,3)
wait_frame=0

class MLPlay:
    def __init__(self, side):
        """
        Constructor

        @param side A string "1P" or "2P" indicates that the `MLPlay` is used by
        which side.
        """
        self.ball_served = False    #未發球
        self.side = side

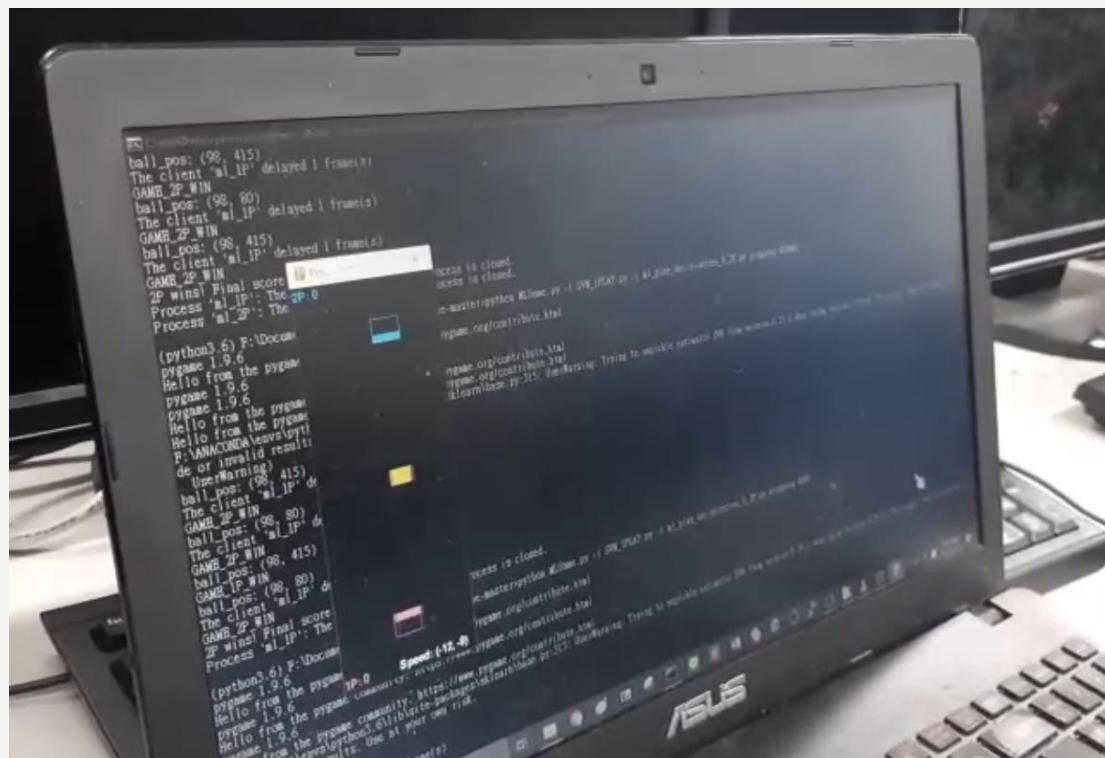
    def update(self, scene_info):
        global wait_frame
        while True:
            if scene_info["status"] != "GAME_ALIVE":    #比出勝負
                return "RESET"    #遊戲重製
            if not self.ball_served:    #如果未發球
                self.ball_served = True
                print("ball_pos:", scene_info["ball"])
                if (ball_served_random==1):
                    return "SERVE_TO_RIGHT"    #往右發球
                else:
                    return "SERVE_TO_LEFT"    #往左發球
            else:
                ball_position_history.append(scene_info["ball"])
                BallPosition=np.asarray(ball_position_history[-1])
                PlatX = np.asarray(scene_info["platform_1P"])[-2])
                Ball_Vx=np.asarray(scene_info["ball_speed"])[-2])
                Ball_Vy=np.asarray(scene_info["ball_speed"])[-1])
                data_x = np.hstack((BallPosition,PlatX,Ball_Vx,Ball_Vy))
```

```
input_data_x = data_x[np.newaxis, :]
move = model.predict(input_data_x)
#move=math.floor(move*10)
print(move)
```

```
if (move < 0):
    return "MOVE_LEFT"
elif (move > 0):
    return "MOVE_RIGHT"
else:
    return "NONE"
return "NONE"
```

```
def reset(self):
    """
    Reset the status
    """
    self.ball_served = False
    ball_served_random=random.randrange(1,3)
```

驗證



Easy 0:3 | C1 win | 最高球速:17

Normal 1:3 | C1 win | 最高球速:24

Hard 0:3 | C1 win | 最高球速:12

https://www.youtube.com/playlist?list=PLbXiz87WNdzaFXTzPJMpj0G-Go6Y64_xn&fbclid=IwARlbZtbi05EbcBkTdAWCttnr0vvA6KghyTOwazqRXvai-8zc9ZGA3etZezw