

作業三Report

Snake

Outline

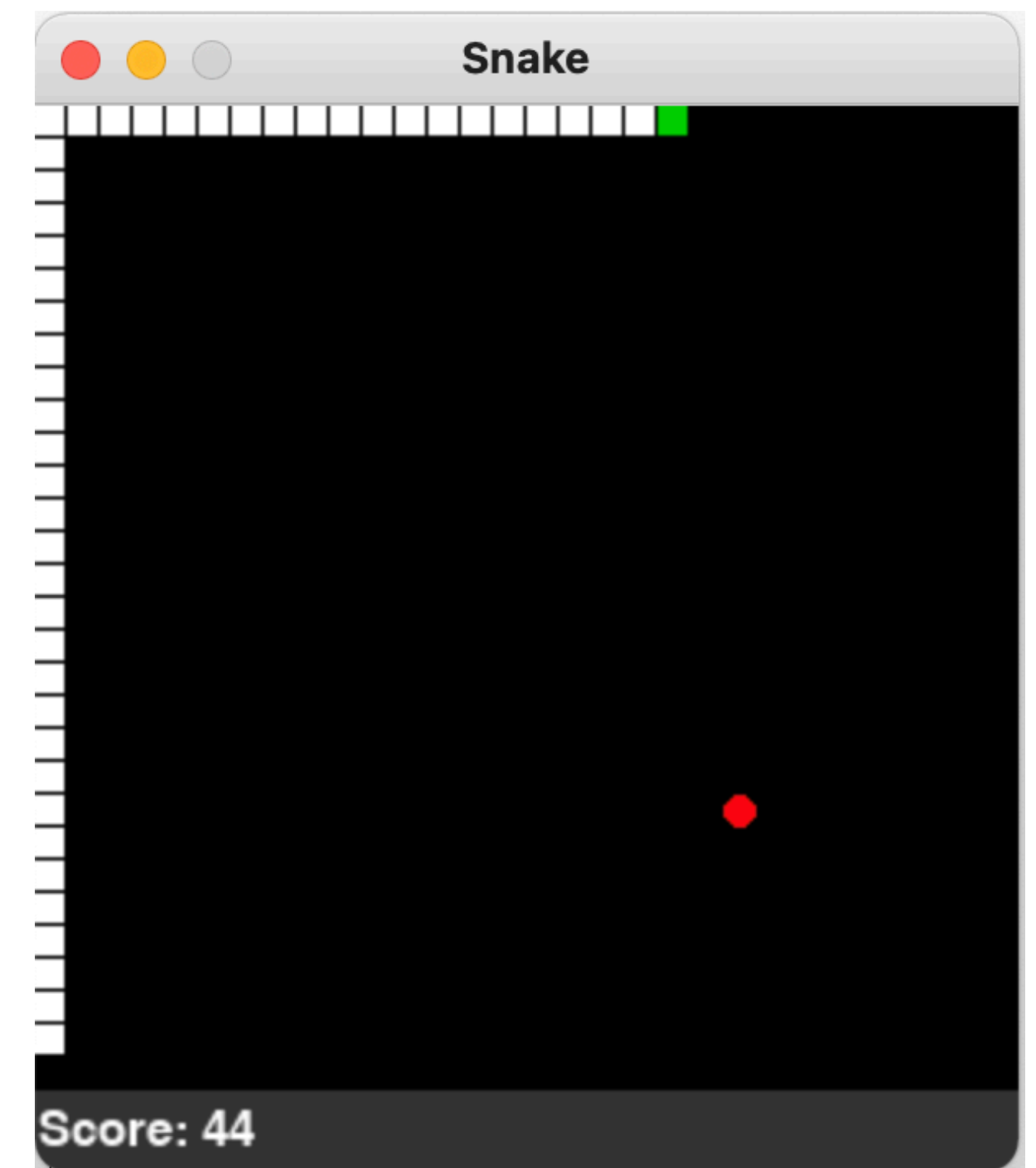
- Introduction
- Method
- Result
- Discussion
- Conclusion

Introduction

Introduction

作業題目及觀察到的現象

- 作業題目 – Snake
- 必須訓練出一個可以令蛇長度超過30的模型。



Introduction

觀察到的現象1-rule的改良-若將蛇的移動路徑改為在場地上重複繞的話效果較佳

- 一開始曾經嘗試讓蛇在最短路徑去吃到下一顆蘋果，但發現效果不彰，訓練出的模型都只能結束在蛇長度30左右，後來嘗試讓蛇在長地上重複繞，效果較佳。
- 結論：我最後使用讓蛇在場地繞的機制

Method

Method

介紹rule

- Rule

1. 讓蛇往上走到(0, 30)。(因為蛇的起始點固定在(40, 30))

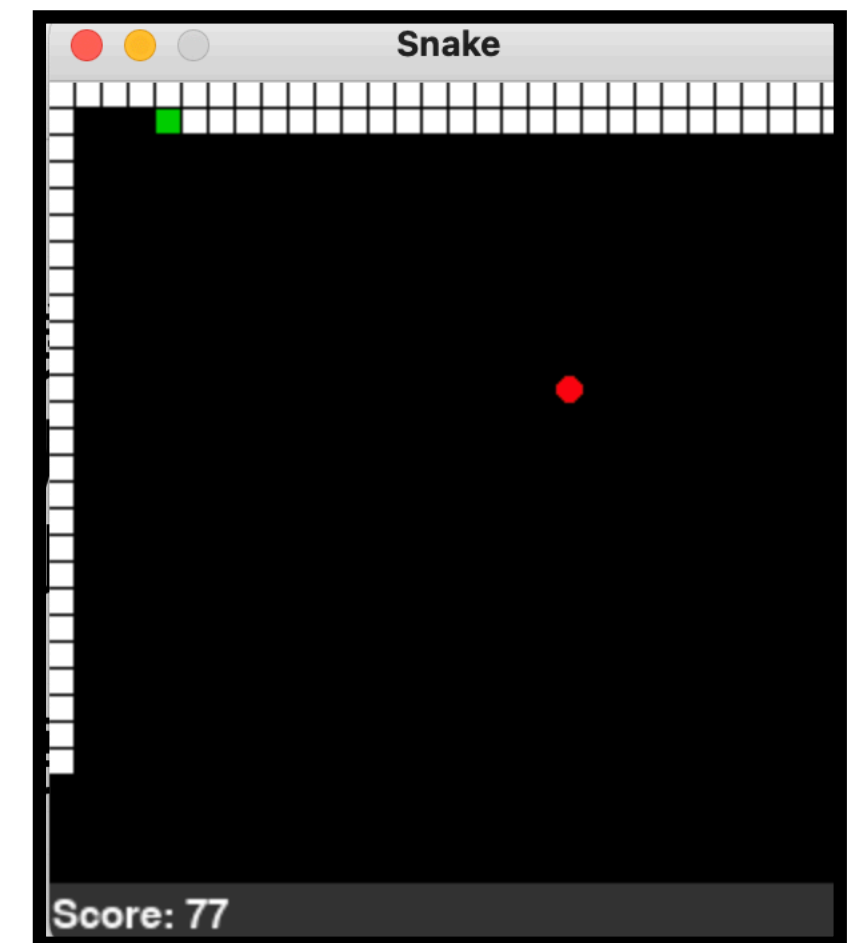
2. 蛇向右走

3. While(蛇沒死){

```
if( (x, y) == (10, 290) ) 向左; if( (x, y) == (0, 290) ) 向上 f( (x, y) == (0, 0) ) 向  
右 ; if(y == 290 && 蛇頭向右) 向下; if(y == 290 && 蛇頭向下) 向左; if(y == 10  
&& 蛇頭像左) 向下; if(y == 10 && 向下) 向右;
```

```
}
```

示意圖



Method

介紹提取之特徵、介紹Pickle檔之搜集

- 提取之特徵-共六個
 1. 蛇頭的x座標
 2. 蛇頭的y座標
 3. 蛇頭的x方向
 4. 蛇頭的y方向
- Pickle檔之搜集：
 - 僅蒐集一組。（目的是為了讓訓練檔overfitting）

Method

介紹所使用的機器學習模型

- 使用的機器學習模型

KNN(k nearest neighbors)

KNN是一種監督式學習，可以做分類或回歸

1. 設定總共要分成幾類

2. 重複計算距離找出k個最相近的特徵

3. 分類: k個特徵投票、回歸: 平均k個特徵

Result

RESULT

如何Train檔案

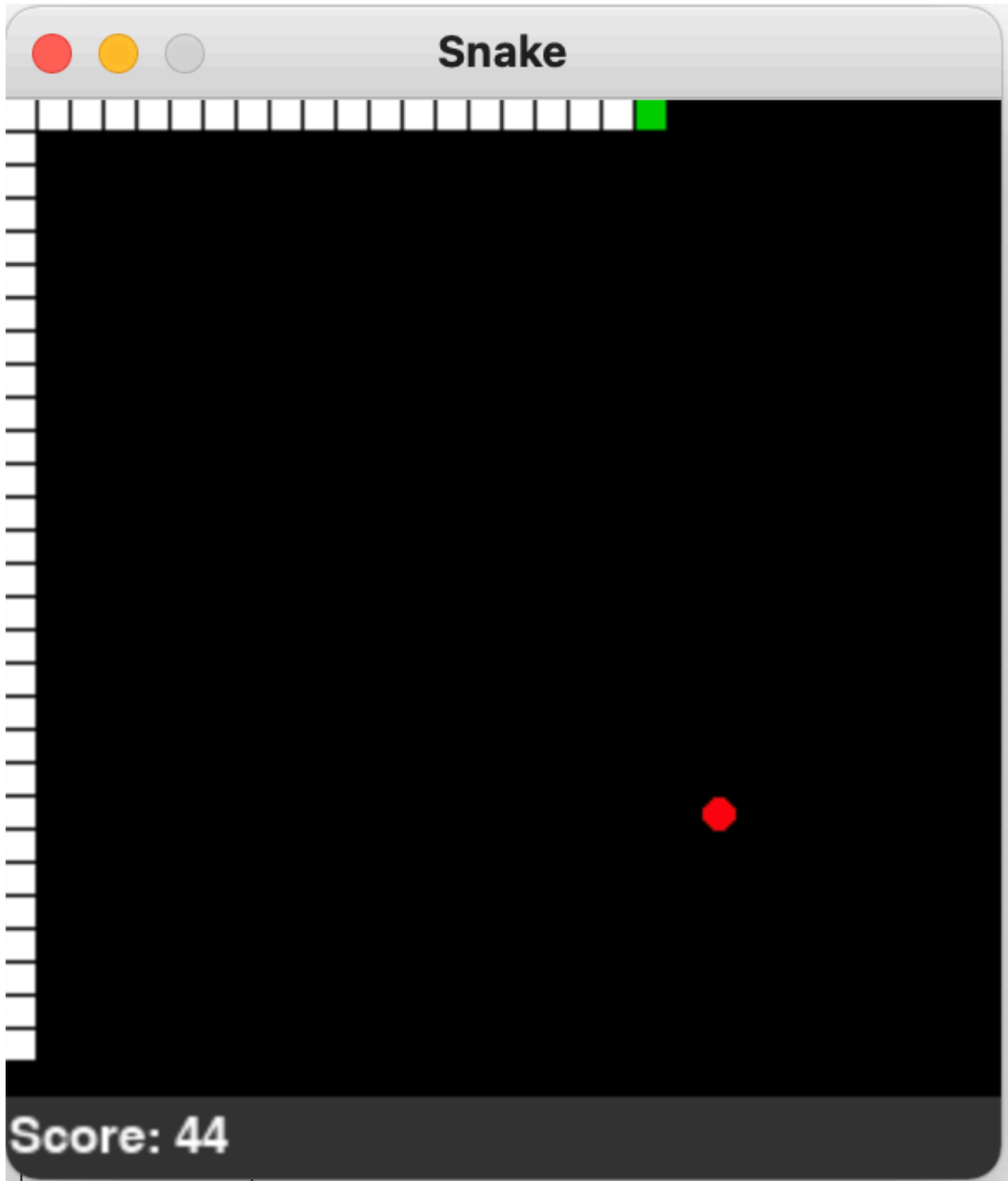
- 只取一個檔案，這個檔案要保證蛇有走過地圖上每一個點至少兩次
- 最後只使用一個檔案無法成功train出模型，因此我使用兩個檔案train。

RESULT

結果

- 實測五次，每一次都有大於30。

	第一局	第二局	第三局	第四局	第五局
是否大於30	是	是	是	是	是



Discussion

Discussion

問題: 訓練data的多寡、data多元性是否足夠，是否影響訓練模型的成效?

- 對data的多元性來說，因為rule每一次的走法都是一樣的，因此並不在意data多元性的問題。
- 以我這一次寫rules的狀況來說，反而不適合使用太多的檔案來train。
 - 因為一個檔案本身就很大了，更多會遇到 1. 訓練時間太久。 2.暫存記憶體空間不足等等的問題。

Conclusion

Conclusion

- 老實說我在這一次的作業中使用了比較“偷吃步”的方法，一開始怎麼寫rules都很難達到穩定的超過長度30。後來我嘗試用手動的模式實際下去玩遊戲，在玩遊戲的過程中就想到可以使用這樣的方法來完成作業。
- 我想這也是我學到的重要的一課，很多東西要實際下去做過解決起來會比較容易。