

1090316 人工智慧概論

Simple 2D Ball Game

By

資工二乙 408262143

林采昕

Date Submitted: April 6

, 2021

Writing

1. Describe your goal formulation and problem formulation.

Goal formulation: 讓我的agent球變最大

Problem formulation:

1. 如何找出目標球，以及達到目標球的位置。
2. 避免碰到邊界。
3. 閃躲大球

2. Describe the search strategy you used, and explain your design concept.

- Search strategy :

從所有球中找出比我小，且利益是最大值得球。抓出他和我的相對位置，來決定前進方向。

若所有比 agent 小的球中的最大值，大於平均+20，則無論此球距離 agent 多遠，都將成為目標球。

- Design concept :

Run function: 使行走方向相反。

Up $\leftarrow \rightarrow$ Down Left $\leftarrow \rightarrow$ Right

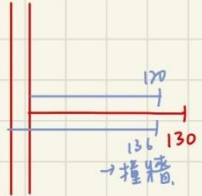
Up_Left $\leftarrow \rightarrow$ Down_Right Up_Right $\leftarrow \rightarrow$ Down_Left

```
12  int run(int action) //往反方向跑
13  {
14      if(action == 1) action = 2;
15      else if(action == 2) action = 1;
16      else if(action == 3) action = 4;
17      else if(action == 4) action = 3;
18      else if(action == 5) action = 8;
19      else if(action == 6) action = 7;
20      else if(action == 7) action = 6;
21      else if(action == 8) action = 5;
22      return action;
23  }
```

將我的座標及目標座標傳進 find function , 來決定前進方向。

```
int find(int xs, int ys, int x, int y) // 跑向目標位置
{
    int action;
    if(xs > x && ys > y) action = UP_LEFT;
    else if(xs > x && ys < y) action = DOWN_LEFT;
    else if(xs < x && ys < y) action = DOWN_RIGHT;
    else if(xs < x && ys > y) action = UP_RIGHT;
    else if(xs == x && ys < y) action = DOWN;
    else if(xs == x && ys > y) action = UP;
    else if(xs > x && ys == y) action = LEFT;
    else if(xs < x && ys == y) action = RIGHT;
    return action;
}
```

```
52     xd = xCoordinate[0] - lastx; // agent X的移動距離
53     yd = yCoordinate[0] - lasty; // agent Y的移動距離
54
55     // 要預留距離, 且右、下為正, 左、上為負
56     // wxd、wyd為躲避邊界與大球 xd、yd為靠近目標球(不用預留距離)
57     int wxd = xd;
58     int wyd = yd;
59
60     xd = xd * (abs(xd) + 1) / 2;
61     yd = yd * (abs(yd) + 1) / 2;
62     wxd += xd;
63     wyd += yd;
64
65     int xdist; // 兩球x軸距離
66     int ydist; // 兩球y軸距離
```



設 agent 離牆 130, 且 x_d 為 15.

當 $x_d = x_d \times \text{abs}(x_d + 1) / 2$
 $\Rightarrow x_d = 120$.

則此時 agent 不會給反方向的力。
 $\Rightarrow x_d$ 為 16 $\rightarrow x_d = x_d \times \text{abs}(x_d + 1) / 2$
 $\Rightarrow x_d = 136$.

\rightarrow 但 $136 > 130 \Rightarrow$ 會撞牆

因此, 將程式改寫
 $\text{int } wxd = x_d$
 $x_d = x_d \times (\text{abs}(x_d) + 1) / 2$
 $wxd += x_d$.

ex, $x_d = 15$
 $wxd = 15$
 $x_d = 15 \times 16 \div 2 = 120$
 $wxd = 120 + 15 = 135$.

或不將 wxd 加上 16 是因為
 x_d 有可能為負,
 若要加 1, 則需判斷正負號.
 但因判斷式有安全距離
 所以少 1 也沒差

判斷目前方向是否朝著大球, 且是否依目前移動速率是否會碰到大球, 若會碰到, 則及時給反方向的力。

```
if(circleRadius[i] > circleRadius[0] && i < 10) // 當半徑比agent大, 並為黑球時
{
    // (wxd * xdist), (wyd * ydist) > 0 代表往大球方向移動
    // abs(wxd) + 1 > abs(xdist), abs(wyd) + 1 > abs(ydist) 代表若不反方向跑則會碰到大球
    if((wxd * xdist > 0 && abs(wxd) + 1 > abs(xdist)) && (wyd * ydist > 0 && abs(wyd) + 1 > abs(ydist))) // 大球在agent附近
    {
        action = find(xCoordinate[0], yCoordinate[0], xCoordinate[i], yCoordinate[i]);
        // 找到目標球的方向
        action = run(action); // 往反方向跑
        flag = 2; // 有大球在附近
    }
}
```

找出綜合判斷位置及半徑後, 利益最高的球變為目標球。

若比 agent 半徑小中的最大球, 半徑比平均大 20, 則無論距離, 直接成為目標球。

因為這樣可以阻止此球變得比我大, 吃掉它後, 自身球也能變大的比較快。

```

else if(circleRadius[i] < circleRadius[0] || i >= 10 )//當半徑比agent小，或為黑球時
{
    ttl += circleRadius[i]; // 所有比agent小的值的總和
    am++; //所有比agent小的值的個數
    if(circleRadius[i] > big)
    {
        big = circleRadius[i];//所有比agent小中最大的值
        tmpbig = i;//所有比agent小中最大的值的球
    }
    int ad = 0; // 利益值
    // x軸平方+y軸平方 = 距離平方
    xdistsq *= xdistsq;
    ydistsq *= ydistsq;
    if( circleRadius[i] > 0) ad = xdistsq + ydistsq + ((200 - circleRadius[i]) * (200 - circleRadius[i])*10);
    // 考慮距離，和半徑大小，但半徑的數值比距離小，所以乘10
    if(ad < minn && ad != 0) //ad為0，代表目標球半徑為0，因此不計入，並找出最值得追的target
    {
        minn = ad;
        locmin = i;
    }
}

```

```

ttl /= am; //算出平均
if(big > ( ttl + 20) )//如果比agent小中的最大球比平均多20，則target換成此球
{
    locmin = tmpbig;
}

```

判斷是否會撞到牆壁

```

//若離邊界太近，則往反方向移動
if((xCoordinate[0] - circleRadius[0] + wxd) < 10 && (yCoordinate[0] - circleRadius[0] + wyd)< 5 && flag != 2)
{
    action = 8;
    flag = 1;
}
else if((xCoordinate[0] + circleRadius[0] + wxd)> 1590 && (yCoordinate[0] - circleRadius[0] + wyd)< 5 && flag != 2)
{
    action = 7;
    flag = 1;
}
else if((xCoordinate[0] - circleRadius[0] + wxd) < 10 && (yCoordinate[0] + circleRadius[0] + wyd) > 895 && flag != 2)
{
    action = 6;
    flag = 1;
}
else if((xCoordinate[0] + circleRadius[0] + wxd)> 1590 && (yCoordinate[0] + circleRadius[0] + wyd) > 895 && flag != 2)
{
    action = 5;
    flag = 1;
}
else if((xCoordinate[0] - circleRadius[0] + wxd) < 10 && flag != 2)
{
    action = 4;
    flag = 1;
}
else if((xCoordinate[0] + circleRadius[0] + wxd)> 1590 && flag != 2)
{
    action = 3;
    flag = 1;
}
else if((yCoordinate[0] - circleRadius[0] + wyd)< 5 && flag != 2)
{
    action = 2;
    flag = 1;
}
else if((yCoordinate[0] + circleRadius[0] + wyd) > 895 && flag != 2)
{
    action = 1;
    flag = 1;
}

```

判斷目標球的位置，並在爆衝前給出反方向的力。

```
int radiusdist;
radiusdist = circleRadius[0] + circleRadius[locmin]; //agent與目標球的半徑距離
radiusdist *= (-1);
if(flag == 0) //若沒碰到球，或沒碰邊界，則朝著target移動
{
    action = find(xCoordinate[0], yCoordinate[0], xCoordinate[locmin], yCoordinate[locmin]);
    //給反方向的力，來阻止碰撞
    if((abs(xCoordinate[0] - xCoordinate[locmin]) + circleRadius[locmin] < xd ) && (abs(yCoordinate[0] - yCoordinate[locmin]) + circleRadius[locmin] < yd ))
    {
        action = run(action); // 往反方向跑
    }
    else if((abs(xCoordinate[0] - xCoordinate[locmin]) + circleRadius[locmin] < xd )) //改變左右方向
    {
        if(action == 3) action = 4;
        else if(action == 4) action = 3;
        else if(action == 5) action = 6;
        else if(action == 6) action = 5;
        else if(action == 7) action = 8;
        else if(action == 8) action = 7;
    }
    else if((abs(yCoordinate[0] - yCoordinate[locmin]) + circleRadius[locmin] < yd )) //改變上下方向
    {
        if(action == 1) action = 2;
        else if(action == 2) action = 1;
        else if(action == 5) action = 7;
        else if(action == 7) action = 5;
        else if(action == 6) action = 8;
        else if(action == 8) action = 6;
    }
}
}
```

```
//將當前位置傳入lastx、lasty
lastx = xCoordinate[0];
lasty = yCoordinate[0];
```

3. Describe the challenges you encountered when designing the agent.

- 不確定座標方向，之後用 freopen 才確定左上(0, 0) 右下是(1599, 899)，也

就是 X 軸方向向右，Y 軸方向向下。

- 原本外接函式寫成 `find(int x, int y)`，把目標座標傳入，直接和 `xCoordinate[0]`、`yCoordinate[0]` 比大小，這樣 `xCoordinate`、`yCoordinate` 沒宣告到。但宣告在外面卻噴錯，最後只好傳入四個值。
- 算出原本要利用加速度公式算出位移量，但算不出來，最後改用反推的方式算出加反向力後的位移。

4. Give two scores from 1 to 10 to evaluate the performance of your teammates in this assignment in terms of Design and Implementation.

陳冠宇 8

林胤軒 7