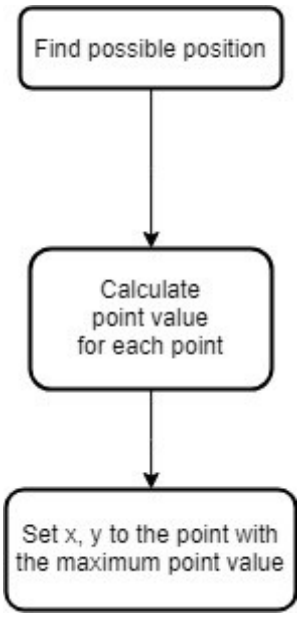
Project 3 Chain Reaction

106062130 邱詠智

1. Flow Chart: (make move function)



2. Detailed Description

a) 整體行為 : (周圍 = 上下左右四個鄰點)

1) 盡量下在不會被同伴 / 敵人影響的地方 (不與同伴/敵人相鄰)
2) 若有可能吃掉周圍的敵人，堆高此點至 Max – 1
3) 與 2 相反(有可能會被吃掉)，儘量不去碰它
4) 周圍的敵人會威脅到自己的時候才引爆(雙方 Max – Record == 1)
5) 當不滿足 2, 3, 4 時，下在白色區域 / 離引爆點較近的我方陣營
白色優先度較高

(所有點都堆至 Max-1/我方陣營周圍皆無敵人/我方陣營會被敵人吃掉)

b) Find possible position :

找出可以下的區域(color == White / color == inputColor)。
根據 point value 從中選出一個作為棋步。
If point value 相等，以白色為優先。

c) point value (priority) :

定義 :

dist = Max[x][y] – Record[x][y] (跟引爆點的距離)
near point 鄰點 = 上下左右 4 個鄰接點

以 dist 的關係可分為 4 種 case

Case 1 : this dist > 1 && this dist <= dist of near point

無法立即引爆，但可以比鄰點更早引爆

Case 2 : this dist > 1 && this dist > dist of near point

無法立即引爆，有可能鄰點比該點更早引爆

Case 3 : this dist == 1 && dist of near point == 1

可以立即引爆，周圍有接近引爆點的鄰點

Case 4 : this dist == 1 && dist of near point > 1

可以立即引爆，周圍沒有接近引爆點的鄰點

再根據鄰點的顏色，決定加權的權重。如下表所示

Case\Color	White	Black	InputColor	enemy
Case 1	0	0	-dist of near point	1000
Case 2				-1000
Case 3				$+\infty$
Case 4				$-\infty$

我的策略中鄰點只關心是否為敵方/我方，因此 White/Black 不影響權重

如果鄰點為我方，下在離引爆點較近的區域，爆炸時一起爆，把更多爆炸後區域染為自己的顏色。(- dist = 離引爆點越近，則權重越大)
如果鄰點為敵人，根據是否能吃掉決定權重，如果能吃就盡量堆高(+1000)，不能就不碰他(-1000)。

如果隨時都能吃掉，那就先放著，等著敵人長肥後一起爆，把更多爆炸後區域染為自己的顏色。

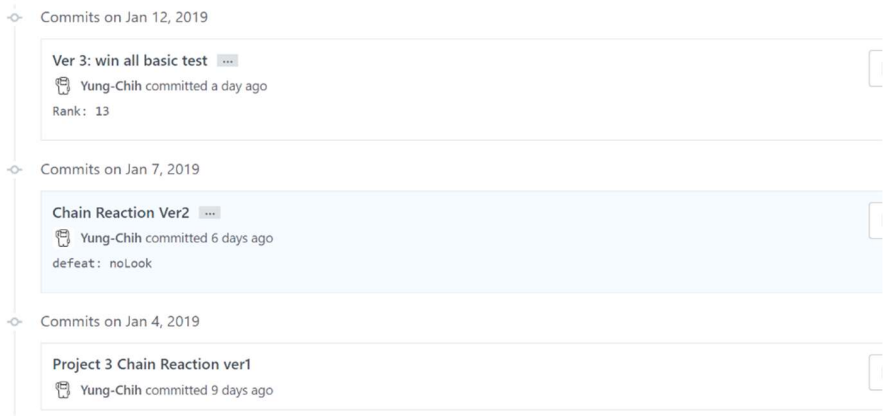
最後比較此點的顏色，若為白色，權重較高、若為 inputColor 則較低。

3. Screen Shot
   a) Partial Code

```
void makeMove(int Record[5][6], int Max[5][6], Color color[5][6], Color inputColor){

    vector<int> pos = this->possible_pos(Record, Max, color, inputColor);

    if( pos.size() == 0 ) return;

    int sel_value = -1e9-10;
    Color sel_c = inputColor;

    for(int i = 0; i < pos.size(); i++ ){
        int now_x = pos[i]/10, now_y = pos[i]%10;
        int v = point_value( now_x, now_y, Record, Max, color, inputColor );

        if( sel_value < v || ( sel_value == v && sel_c == inputColor && color[now_x][now_y] == White ) ){
            this->x = now_x;
            this->y = now_y;
            sel_value = v;
            sel_c = color[now_x][now_y];
        }

    }
}
int getX(){
    return x;
}
int getY(){
    return y;
}
```

```
bool valid( int x, int y ){
    return( x >= 0 && x < 5 && y >= 0 && y < 6 );
}

int point_value(int x, int y, int Record[5][6], int Max[5][6], Color color[5][6], Color inputColor ){
    int dist = Max[x][y] - Record[x][y];

    int value = 0;
    for(int i = 0; i<4; i++){
        int nx = x + dir_x[i], ny = y + dir_y[i];
        if( !valid( nx, ny ) ) continue;
        if( color[nx][ny] == White || color[nx][ny] == Black ) continue;

        int nd = Max[nx][ny] - Record[nx][ny];

        if( color[nx][ny] != inputColor){
            /// if it can eat enemy increase priority
            /// if not, decrease priority
            if( dist == 1 ){
                if( nd == 1 ) return 1e9;
                else          return -1e8;
            }
            else{
                if( dist <= nd ) value += 1000;
                else             value -= 1000;
            }
        }
        else{
            value -= nd;
        }
    }
    if( dist == 1 )
        return -1e7;
    else
        return ( color[x][y] == White ) ? value - dist : value - 10*dist;
}
```

b) GitHub History



c) TA's AI vs My AI:

by yourself

StudentId	randomMove	noLook	heithoff	rlawrenc
106062130	Pass	Pass	Pass	Pass

d) Rank 11

106072123	4	66	44	
106062119	5	73	176	
1071167s	6	242	54	Challenge
105021217	7	122	157	Challenge
106062203	8	161	21	Challenge
106081005	9	70	26	Challenge
105021121	10	93	70	Challenge
106062130	11	32	52	
105020011	12	134	110	