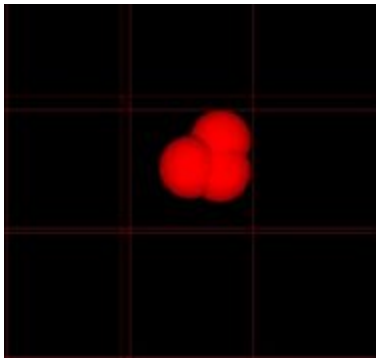
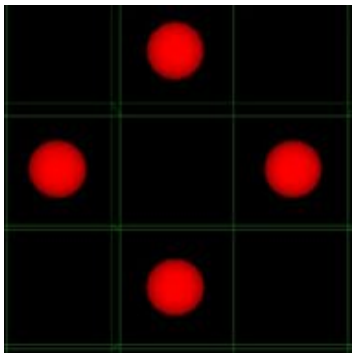
Problem Description:

1. The game starts with an empty board with 8x8 cells.
2. It will be a 2 player game, let one player be red and one be green.
3. Players take turns to place their orbs in a cell. One can add an orb in an empty cell or in a cell with his/her orbs. (e.g. a green player can place a green orb at an empty cell or in a cell with green orbs. Green orb cannot be placed in a cell containing red orb.)
4. When a cell has 3 orbs, adding a fourth one makes it explode, which causes 4 orbs to be added to adjacent 4 cells (up, down, left and right cell). If those 4 cells contain orbs of different color, they will convert to the color of exploded orb's.
5. If cell A has 3 orbs already, and one of adjacent cells of A gets exploded, then one orb is added to A, which causes A to explode also. This may result to a chain reaction.
6. The player to claim all of the orbs will be the winner.
7. If the cell is located at one corner or in the edge, then it won't have 4 adjacent cells. Corner cells have 2 adjacent cells, cells located in the edge have 3 adjacent cells, so a corner cell will explode if it has 2 orbs, and an edge cell will explode when 3rd orb is added to it.

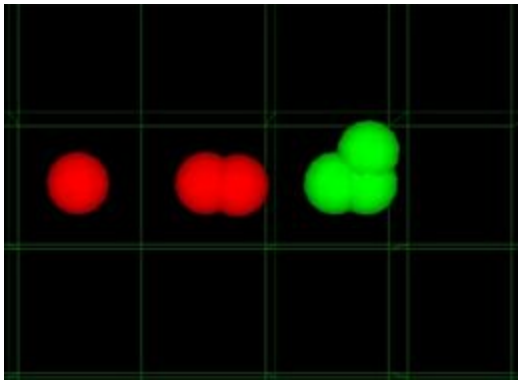
Figures:

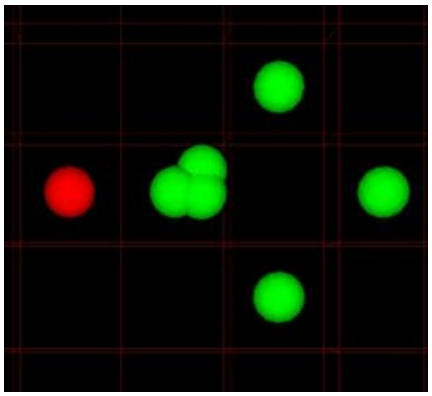
1. Adding a red orb to the cell with 3 red orbs make it explode, 4 orbs are added to 4 adjacent cells.



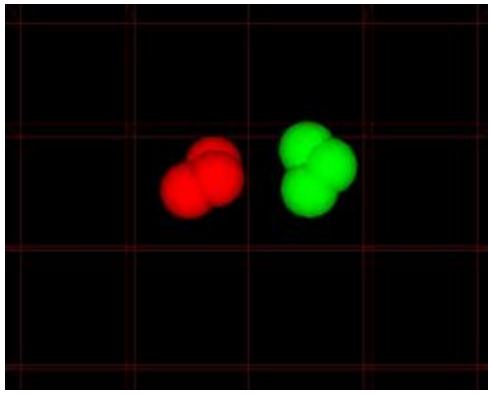


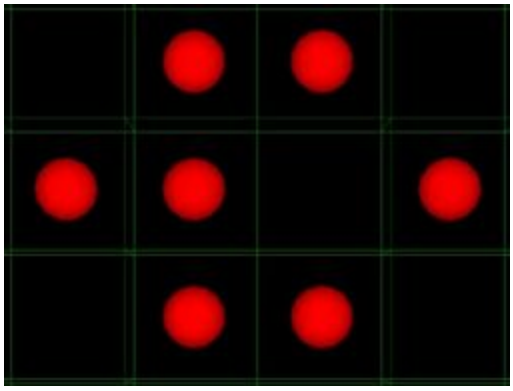
2. After adding a green orb to the cell with 3 green orbs, it explodes, and 2 red orbs in the left adjacent cell converts to green due to green orb explosion.





3. Adding a red orb to the cell with 3 orbs causes it to explode, which adds 1 orb to adjacent cell with 3 green orbs (and converts all to red). So, it becomes a cell with 4 red orbs, and so it also explodes. (Chain Reaction) Here all the orbs has turned to red, so the game's over and red wins.





N.B. : This problem is designed from the game "Chain Reaction" . You can get it from Google Play for more clarification.

https://play.google.com/store/apps/details?id=com.BuddyMattEnt.ChainReaction&hl=en

Problem Specification:

For File Input Mode: (code: aicontest_file.py)

- Participants will be given a assigned color('R'/'G') for the match through command line argument.
- Participants will read from the file "shared_file.txt". The first line of the file declares the color of the player to make the current move. If the first line matches with the assigned color of the participant, they can make the move.

- The following lines of the file will contain current condition of the grid. There will be 8 rows, each row has 8 strings. Each string indicates the current state of the corresponding cell in the grid. If the value of the string is “No”, then the grid is empty. Otherwise, it will contain 2 characters, first one denoting color of the orb (R/G), and 2nd one tells the number of orbs. Like, if “G2” is found on the 2nd string of 3rd row, that means cell[3, 2] contains 2 green balls.
- To make the move participants will write '0' in the first line of the file "shared_file.txt". In the second line they will write x and y are coordinates for placing next orb in space separated manner.

Here is a state of the **shared_file.txt** after **aicontest_file.py** file has written to the **R** player:

```
R
No G1 No No R1 R1 G2 No
No G1 R1 G1 R3 No R2 R2
G1 R1 G1 G1 R1 G1 No No
No G1 No R1 R1 No G1 No
G1 No No No No G1 G2 R1
R1 G1 R1 R1 G1 No R2 R1
No R2 No G2 No R1 R1 No
G1 G2 No G1 No G2 No No
```

For Console Input Mode: (code: aicontest.py)

- Participants will receive “start” command from the console. (Can be read through scanf() or equivalent functions).
- After that participants will receive the current situation of the grid from the console. There will be 8 rows, each row has 8 strings. Each string indicates the current state of the corresponding cell in the grid. If the value of the string is “No”, then the grid is empty. Otherwise, it will contain 2 characters, first one denoting color of the orb (R/G), and 2nd one tells the number of orbs. Like, if “G2” is found on the 2nd string of 3rd row, that means cell[3, 2] contains 2 green balls.
- Participants can just print coordinates in console in a space separated manner (In C, printf(“%d %d”, x, y) will work, where x and y are coordinates for placing next orb).

- For console output mode participants need to change the following lines (line 327, line 329) in the aicontest.py to add the command to run their code accordingly.

```
p1 = subprocess.Popen(['python3', 'player_code.py', 'R'],
stdout=subprocess.PIPE, stdin=subprocess.PIPE,
                          universal_newlines=True, bufsize=1)
```

```
p2 = subprocess.Popen(['python3', 'player_code.py', 'G'],
stdout=subprocess.PIPE, stdin=subprocess.PIPE,
                          universal_newlines=True, bufsize=1)
```

for example if the participants writes code in C/C++ and generates output code named a.out the lines should be -

```
    p1 = subprocess.Popen(['./a.out', 'R'], stdout=subprocess.PIPE,
stdin=subprocess.PIPE,
                          universal_newlines=True, bufsize=1)
    p2 = subprocess.Popen(['./a.out', 'G'], stdout=subprocess.PIPE,
stdin=subprocess.PIPE,
                          universal_newlines=True, bufsize=1)
```

- Participants can use any one of the above modes.
- Participants need to make the move in reasonable amount of time.
- Producing invalid move will disqualify the participant.
- Participants must not make any other changes in the files **aicontest.py** or **aicontest_file.py**
- Participants only need to write the **player_code** using the programming language of their preference.
- Participants can look into the given sample player codes for further clarification.

Instruction for running:

- Download the sample code from https://github.com/JakeShammo/CSE_Fest_AI_Contest repository
- Run the **aicontest.py/aicontest_file.py** using **python3** adding graphics speed as a command line argument. Here graphics speed denotes the speed of moves shown in the ui. You can try with different speed for your convenience. [**python3 aicontest.py 1000**]. You may have to install package **"numpy", "PyOpenGl" (>=3.0), "Pygame" (>=1.9.0)** to run this script (can be installed using **pip**) .
- Run the **player_code.py** using python3 adding “R/G” as a commandline argument. [**python3 player_code.py R]**

Before running the **aicontest.py** or **aicontest_file.py** install required packages using:

```
pip install -r Requirements.txt
```

For console input output mode run:

```
python aicontest.py 1000
```

For file input output mode run:

```
python aicontest_file.py 1000
python player_code_file.py R
python player_code_file.py G
```

Video Demo:

Here goes a demo match using our sample bot: https://goo.gl/LBsnnD