AVISHKAR'22

CODE WARRIORS 2K22

BROUGHT TO YOU BY THE AVISHKAR TEAM

FOR B.TECH. 2ND AND 3RD YEAR
AND MCA 1ST AND 2ND YEAR

EVENT COORDINATORS

KARIMULLA MOHAMMAD	DIVYANSH UPADHYAY
7879324647	8630187335
karimulla.mohammad@mnnit.ac.in	divyansh.upadhyay@mnnit.ac.in

EVENT DESCRIPTION

• Code Warriors is CyberQuest's flagship event under the Artificial Intelligence category. In this event, there will be only one problem statement challenge – write a bot for the game. The game will be a two-player game and will be played by bots against each other. Your bot has to beat the other bot while adhering to the rules of the game.

• Teams will compete each other in multiple knockout rounds which will be held during Avishkar. Each round might have multiple games to ensure fair opportunity to both bots to play as player 1. The bot that wins the most rounds will win. In case of a tie, secondary metrics like moves needed to win may be used at the discretion of the coordinators.

Tips for the event:

• For beginners, starter agents have been provided which at each point makes a random valid move on the arena.

• Additionally, you will also be judged on the basis of code structure and reasoning of the models.

• For any problems, queries, questions, or help at any stage, contact seniors or reach out to peers.

RULES AND REGULATIONS

- This is a team event of maximum 2 members. You can still work individually if you want. Team members can be from any branch.

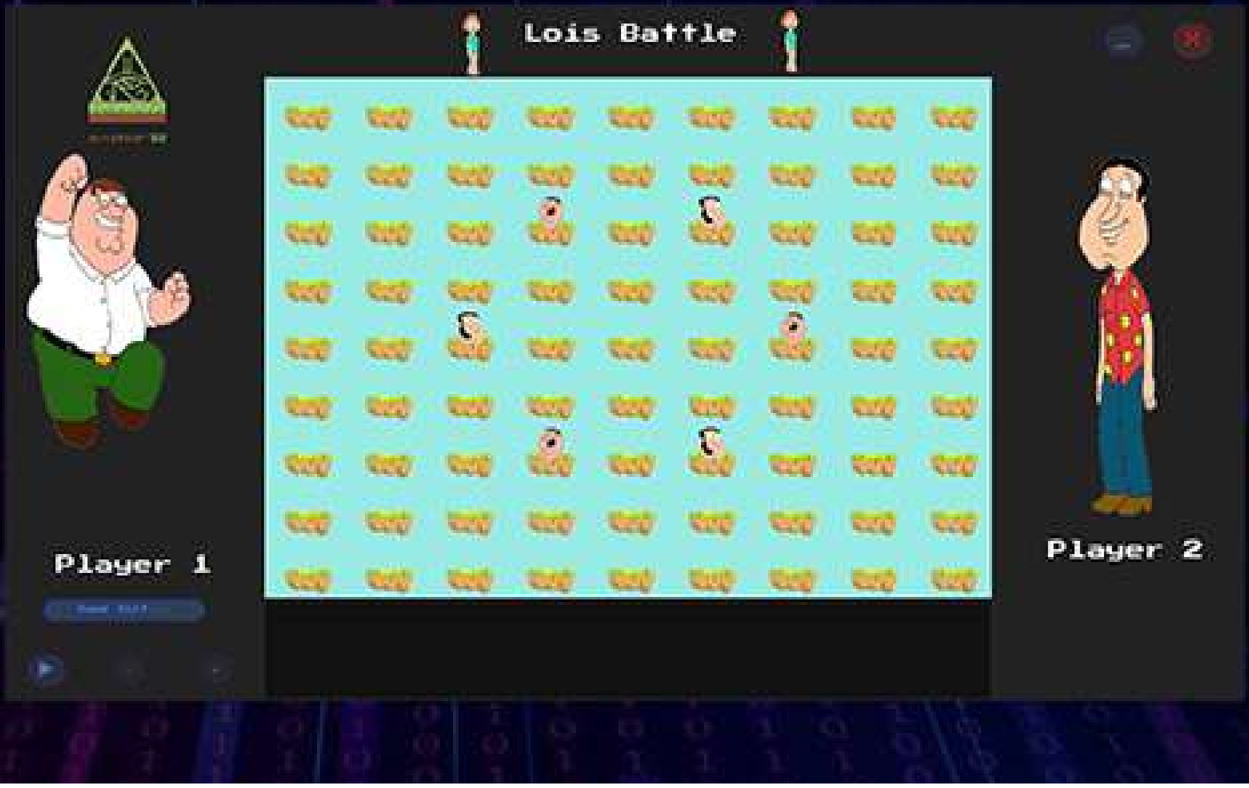
- The only prerequisite is that at-least one member of the team should be from the CS/IT branch.

- Cross year teams are not allowed.

- The decision of the judges will be final and binding.

- ANY FORM OF CODE PLAGIARISM WILL NOT BE TOLERATED.

ARENA



TERMINOLOGY

- Arena: The main board consists of a 9x9 grid.

- Players: There are two players and each player has 3 coins with initial spawn position fixed as shown in the image.

- History Panel: The history panel on the middle-bottom shows the history of moves performed by both the players.

GAME RULES

- Player 1 plays the first turn and in each turn a player can move one of its coins to its adjacent location.

- In total both player 1 and player 2 combined can visit a cell for 3 times after that no player can move to that cell.

- Each cell in the grid has a state. Each cell starts with state 0 and the state increases when the player leaves the cell. No player can visit the cell once its state reaches 3.

- Each player will get its turn alternately and failing to make a move means that player loses the round.

- No player can move the same coin in his/her two consecutive turns.

- 2 rounds will be played between two bots. The player will get swapped in the next round i.e if a bot gets player 1 in round 1 it will get player 2 in round 2 and vice versa.

- In case of a tie, secondary metrics like moves taken to win may be used at the discretion of the coordinators.

INPUT

- Sample input:

```
9 9 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
2 3 6 3 4 6
2 5 6 5 4 2
-1 -1
```

- The first row contains three integers, the first two being dimensions of the board (9,9) which are fixed. The third integer is the player whose turn it is to move. 1<=player<=2

- The next 9 rows and 9 columns contain the state of each cell. 0<=state<=3

- The next line contains six integers, the x,y coordinates of coins of player 1. coin1x, coin1y, coin2x, coin2y, coin3x, coin3y.

- Similarly the next line contains six integers, the x,y coordinates of player 2.

OUTPUT

- Output 3 integers: c x y
c: coin number which you want to move 1<=c<=3.
x y: the new x,y coordinate where you want to move the coin 0<=x,y<=8

- Sample:

```
1 2 7
```

INSTRUCTIONS FOR RUNNING THE ARENA

- Before running the build file on the system, ensure that you have Java 16 installed.

- Download the arena from: https://bit.ly/3zqtgHU

- Extract the zip file.It contains the jar file for the Arena

- Either open the jar file by double clicking it or running the jar file running the command "java -jar LoisBattle.jar" in terminal

- You can also find the starter agents for the event in the " starter agents " folder

- For doubts, you can contact any of the event coordinators

DESIGNED BY TEAM
MEDIA HOUSE OF MNNIT

MEDIA
HOUSE
MNNIT