**BATTLE CHESS**

**Problem Statement :->**

A chess game with a little twist in rules and gameplay using concepts of Object Oriented Programming.

**Description of project :->**

The project is based on a chess game. The rules and gameplay of the game is quite different from the normal chess. Each player has its own restricted area for movement of the pieces. There is a functionality which allows the pieces to shoot the opponents and destroy them. Each piece has different intensity to shoot and they can shoot in different directions.

playerclass

+name : char

+\*mypiece : blueclass/redclass

(using template)

playerclass(char,template);

+void turn();

+void display();

bluepieces/redpieces

+\*piece[16] : piececlass

bluepieces() / redpieces() : constructor

+void display();

Association

1

1

piececlass

+x : int

+y : int

+player : int

piececlass(int ,int ,int);

+void put(int ,int);

+void location(in , int);

+void shoot();

+int check(int,int);

+virtual void display()=0;

+int validity();

bluepieces/redpieces

+\*piece[16] : piececlass

+score : int

bluepieces() / redpieces() : constructor

+void display();

Aggregation

1

M

|  |
| --- |
| piececlass |
| +x : int  +y : int  +player : int |
| piececlass(int ,int ,int);  +void put();  +void move();  +void shoot();  +int check();  **+*virtual* void display()=0;**  **+virtual int validity()=0;** |

|  |
| --- |
| Pawnclass |
| +void display();  +int validity(); |

|  |
| --- |
| bishopclass |
| +void display();  +int validity(); |

|  |
| --- |
| kingclass |
| +void display();  +int validity(); |

|  |
| --- |
| queenclass |
| +void display();  +int validity(); |

|  |
| --- |
| knightclass |
| +void display();  +int validity(); |

|  |
| --- |
| rookclass |
| +void display();  +int validity(); |