**BlueRose Technologies-Cloud BU**

**Hackathon 1(1st April 2023)**

**Title: Battleship**

Table of Contents:

* Introduction
* Scope of Document
* Technical Desgin & Flow

## Overview:

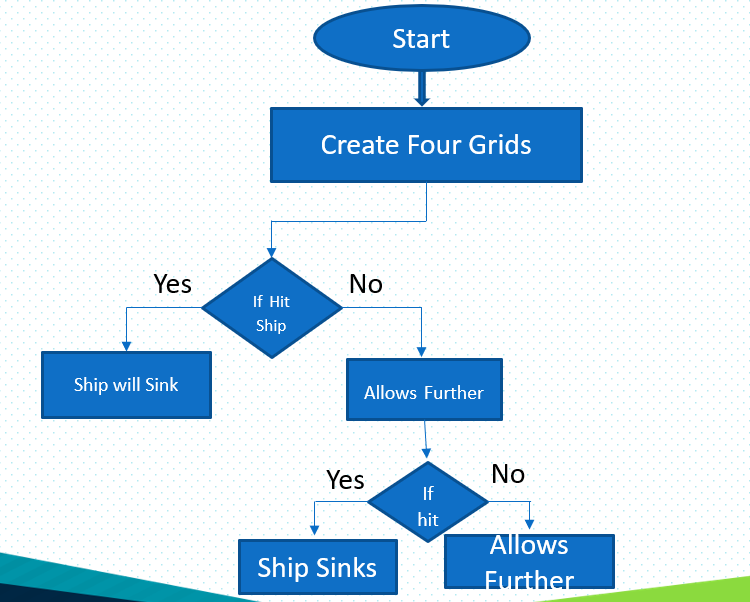
This document represents the technical design for Hackathon-1. This document will only specify the Functional & technical overview of the battleship game done by Python Code in Visual studio tool.

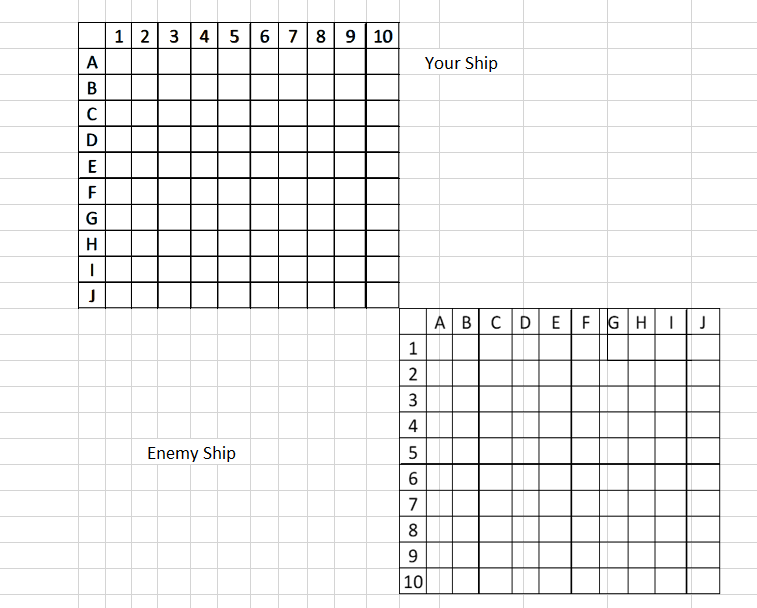
# Introduction**:**

Battleship game is a strategy type guessing game for two players. It is played on ruled grids on which each player’s fleet of warship are marked. Players alternate calling “shot” at the other player’s ship and the objective of the game is to destroy the opposing player’s fleet.



(A map of one player’s ships and the hits against them from a game in progress. The grey boxes are the ships placed by the player, and the cross marks show the squares that their opponent has fired upon. The player would be tracking the success of their own shots in a separate grid)





* The Game is played on four grids, two for each player. The grids are typica square – usually 10\*10 and the individual squares in the grid identified by letters or number.
* On one grid the player arranges ships and records the shots by the opponent. On the other grid, the player records their own shots.
* Before play begins, each player secretly arranges their ships on their primary grid. Each ship occupies several consecutive squares on the grid, arranged either horizontally or vertically.
* The number of squares for each ship is determined by the type of ship. The ships cannot overlap (i.e. only one ship can occupy any given square in the grid).
* The types and numbers of ships allowed are the same for each player. These may vary depending on the rules.
* The ship should be hidden from players sight, and it’s not allowed to see each other’ pieces.
* The game is a discovery game which players need to discover their opponents ship positions.

CODE1:

To define maps that is used in game :

'''#making ship rows for player 1

ship\_row1 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row2 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row3 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row4 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row5 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row6 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row7 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row8 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row9 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

ship\_row10=[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

#making attack map rows for player 1

map\_row1 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row2 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row3 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row4 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row5 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row6 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row7 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row8 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row9 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

map\_row10=[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

#making ship rows for player 2

two\_ship\_row1 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row2 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row3 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row4 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row5 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row6 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row7 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row8 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row9 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_ship\_row10=[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

#making attack map rows for player 2

two\_map\_row1 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row2 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row3 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row4 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row5 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row6 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row7 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row8 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row9 =[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

two\_map\_row10=[" |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "," |\_| "]

player1shipmap1=[ship\_row1,ship\_row2,ship\_row3,ship\_row4,ship\_row5,ship\_row6,ship\_row7,ship\_row8,ship\_row9,ship\_row10]

player1map1=[map\_row1,map\_row2,map\_row3,map\_row4,map\_row5,map\_row6,map\_row7,map\_row8,map\_row9,map\_row10]

player2shipmap=[two\_ship\_row1,two\_ship\_row2,two\_ship\_row3,two\_ship\_row4,two\_ship\_row5,two\_ship\_row6,two\_ship\_row7,two\_ship\_row8,two\_ship\_row9,two\_ship\_row10]

player2map=[two\_map\_row1,two\_map\_row2,two\_map\_row3,two\_map\_row4,two\_map\_row5,two\_map\_row6,two\_map\_row7,two\_map\_row8,two\_map\_row9,two\_map\_row10]

CODE2:

INTERACTING WITH THE USER:

while totalshipsremaining>0:

print("\t\t\t\tPlease dont place a ship where already a ship is present")

print("\t\tPlace your ships")

choice=int(input("Enter choice\n1=AircraftCarrier\n2=Battleship\n3=Cruiser\n4=Destroyer\n5=Submarine\n"))

#to place aircraft ship

if choice==1 and AircraftCarrier>0:

row=int(input("Enter the rownumber in the map where you want to place the Aircraft carrier\n It takes 5 boxes \nNote:from this point it faces right\n"))

col=int(input("Enter the Column number in the map where you want to place the Aircraft carrier\n It takes 5 boxes\nNote:from this point it faces down\n"))

alignment=input("Enter how you want to place it (horizontal/vertical)\n")

totalshipsremaining=totalshipsremaining-1

if alignment=="horizontal" and col<=6:

for i in range(0,5):

#underbeta testing - the following 3 lines

#if player1shipmap1[row-1][col+i-1]==" A " or player1shipmap1[row-1][col+i-1]==" B " or player1shipmap1[row-1][col+i-1]==" C " or player1shipmap1[row-1][col+i-1]==" D " or player1shipmap1[row-1][col+i-1]==" S ":

#print("a Ship is already present")

#else:

player1shipmap1[row-1][col+i-1]=" A "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

elif alignment=="vertical" and row<=6:

for i in range(0,5):

player1shipmap1[row+i-1][col-1]=" A "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

else:

print("No space to place ship")

AircraftCarrier=AircraftCarrier-1

printmap(player1shipmap1)

#to place Battleship

elif choice==2 and Battleship>0 :

row=int(input("Enter the rownumber in the map where you want to place the Battleship\nIt takes 4 boxes \nNote:from this point it faces right\n"))

col=int(input("Enter the Column number in the map where you want to place the Battleship\nIt takes 4 boxes\nNote:from this point it faces down\n"))

alignment=input("Enter how you want to place it (horizontal/vertical)\n")

totalshipsremaining=totalshipsremaining-1

if alignment=="horizontal" and col<=5:

for i in range(0,4):

player1shipmap1[row-1][col+i-1]=" B "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

elif alignment=="vertical" and row<=5:

for i in range(0,4):

player1shipmap1[row+i-1][col-1]=" B "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

else:

print("No space to place ship")

Battleship=Battleship-1

printmap(player1shipmap1)

#to Place Cruiser

elif choice==3 and Cruiser>0 :

row=int(input("Enter the rownumber in the map where you want to place the Cruiser\nIt takes 3 boxes \nNote:from this point it faces right\n"))

col=int(input("Enter the Column number in the map where you want to place the Cruiser\nIt takes 3 boxes\nNote:from this point it faces down\n"))

alignment=input("Enter how you want to place it (horizontal/vertical)\n")

totalshipsremaining=totalshipsremaining-1

if alignment=="horizontal" and col<=4:

for i in range(0,3):

player1shipmap1[row-1][col+i-1]=" C "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

elif alignment=="vertical" and row<=4:

for i in range(0,3):

player1shipmap1[row+i-1][col-1]=" C "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

else:

print("No space to place ship")

Cruiser=Cruiser-1

printmap(player1shipmap1)

#to place Destroyer

elif choice==4 and Destroyer>0:

row=int(input("Enter the rownumber in the map where you want to place the Destroyer\nIt takes 2 boxes \nNote:from this point it faces right\n"))

col=int(input("Enter the Column number in the map where you want to place the Destroyer\nIt takes 2 boxes\nNote:from this point it faces down\n"))

alignment=input("Enter how you want to place it (horizontal/vertical)\n")

totalshipsremaining=totalshipsremaining-1

if alignment=="horizontal" and col<=3:

for i in range(0,2):

player1shipmap1[row-1][col+i-1]=" D "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

elif alignment=="vertical" and row<=3:

for i in range(0,2):

player1shipmap1[row+i-1][col-1]=" D "

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

else:

print("No space to place ship")

Destroyer=Destroyer-1

printmap(player1shipmap1)

#to place Submarine

elif choice==5 and Submarine>0:

row=int(input("Enter the rownumber in the map where you want to place the Submarine\nIt takes 1 box \n"))

col=int(input("Enter the Column number in the map where you want to place the Submarine\nIt takes 1 box\n"))

player1shipmap1[row-1][col-1]=" S "

totalshipsremaining=totalshipsremaining-1

print(f"\t\t\t\t\tRemaining ships={totalshipsremaining}")

printmap(player1shipmap1)

Submarine=Submarine-1

#no ships available

else:

print("No ships available or Invalid entry")