Python Code Design and Implementation

# Design:

I have decided to create a game of Battleships (player vs CPU). The application will create a board within the console.

A screenshot of a computer screen

Description automatically generated

# Implementation:

Steps for programming

1. Output view of the board and store each tile in an array.
2. Allow the user to select difficulty levels.
3. Generate different types of ships with different lengths and random positions.
4. Allow users to make guesses on where ships are located.
5. Show any guesses as right or wrong on an updated board each turn.
6. Calculate when the user wins/loses.

Imperative Pseudocode:

A screenshot of a computer

Description automatically generated

# First Iteration of code:

|  |
| --- |
| print(" 0 1 2 3 4 5 6 7 8 9")  T = [  ["0|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["1|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["2|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["3|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["4|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["5|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["6|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["7|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["8|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["9|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"]  ]  #Generates a random co-ord for the Aircraft Carrier to go  UpOrDown = random.randint(0, 1)  if UpOrDown == 0:  aircraftpositionx = random.randint(0, 5)  aircraftpositiony = random.randint(0, 9)  #Maps the Aircraft Carrier to the co-ords given  AircraftTiles = [[aircraftpositionx, aircraftpositiony],  [aircraftpositionx + 1, aircraftpositiony],  [aircraftpositionx + 2, aircraftpositiony],  [aircraftpositionx + 3, aircraftpositiony],  [aircraftpositionx + 4, aircraftpositiony]]  else:  aircraftpositionx = random.randint(0, 9)  aircraftpositiony = random.randint(0, 5)  #Maps the Aircraft Carrier to the co-ords given  AircraftTiles = [[aircraftpositionx, aircraftpositiony],  [aircraftpositionx, aircraftpositiony + 1],  [aircraftpositionx, aircraftpositiony + 2],  [aircraftpositionx, aircraftpositiony + 3],  [aircraftpositionx, aircraftpositiony + 4]]  #Adds all the tiles used to a used tile list  UsedTiles = []  UsedTiles.extend(AircraftTiles)  #Frigate Generation  FrigateTiles = []  #Checks theres no boat in the current position  while not FrigateTiles or any(x in UsedTiles for x in FrigateTiles):  if UpOrDown == 0:  FrigatePositionX = random.randint(0, 6)  FrigatePositionY = random.randint(0, 9)  FrigateTiles = [[FrigatePositionX, FrigatePositionY],  [FrigatePositionX + 1, FrigatePositionY],  [FrigatePositionX + 2, FrigatePositionY],  [FrigatePositionX + 3, FrigatePositionY]]  else:  FrigatePositionX = random.randint(0, 9)  FrigatePositionY = random.randint(0, 6)  FrigateTiles = [[FrigatePositionX, FrigatePositionX],  [FrigatePositionX, FrigatePositionY + 1],  [FrigatePositionX, FrigatePositionY + 2],  [FrigatePositionX, FrigatePositionY + 3]]  UsedTiles.extend(FrigateTiles)  #Cruiser Generation  CruiserTiles = []  while not CruiserTiles or any(x in UsedTiles for x in CruiserTiles):  if UpOrDown == 0:  CruiserPositionX = random.randint(0, 7)  CruiserPositionY = random.randint(0, 9)  CruiserTiles = [[CruiserPositionX, CruiserPositionY],  [CruiserPositionX + 1, CruiserPositionY],  [CruiserPositionX + 2, CruiserPositionY]]  else:  CruiserPositionX = random.randint(0, 9)  CruiserPositionY = random.randint(0, 7)  CruiserTiles = [[CruiserPositionX, CruiserPositionY],  [CruiserPositionX, CruiserPositionY + 1],  [CruiserPositionX, CruiserPositionY + 2]]  UsedTiles = UsedTiles + CruiserTiles  #Destroyer Generation  DestroyerTiles = []  while not DestroyerTiles or any(x in UsedTiles for x in DestroyerTiles):  if UpOrDown == 0:  DestroyerPositionX = random.randint(0, 8)  DestroyerPositionY = random.randint(0, 9)  DestroyerTiles = [[DestroyerPositionX, DestroyerPositionY],  [DestroyerPositionX + 1, DestroyerPositionY]]  else:  DestroyerPositionX = random.randint(0, 9)  DestroyerPositionY = random.randint(0, 8)  DestroyerTiles = [[DestroyerPositionX, DestroyerPositionY],  [DestroyerPositionX, DestroyerPositionY + 1]]  UsedTiles = UsedTiles + DestroyerTiles  #Destroyer Generation  Destroyer2Tiles = []  while not Destroyer2Tiles or any(x in UsedTiles for x in Destroyer2Tiles):  if UpOrDown == 0:  Destroyer2PositionX = random.randint(0, 8)  Destroyer2PositionY = random.randint(0, 9)  Destroyer2Tiles = [[Destroyer2PositionX, Destroyer2PositionY],  [Destroyer2PositionX + 1, Destroyer2PositionY]]  else:  Destroyer2PositionX = random.randint(0, 9)  Destroyer2PositionY = random.randint(0, 8)  Destroyer2Tiles = [[Destroyer2PositionX, Destroyer2PositionY],  [Destroyer2PositionX, Destroyer2PositionY + 1]]  UsedTiles = UsedTiles + Destroyer2Tiles  hits = 0  for i in range(len(T)):  for j in range(len(T[i])):  print(T[i][j], end=' ')  print()  while hits <= 15:  xguess = int(input("Enter row co-ordinate: "))  yguess = int(input("Enter column co-ordinate: "))  guessed = [[xguess, yguess]]  if any(x in UsedTiles for x in guessed):  print("Hit!")  hits = hits + 1  if yguess == 9:  T[xguess][yguess] = "|XX|\n"  else:  T[xguess][yguess] = "|XX"  else:  print("Miss!")  if yguess == 9:  T[xguess][yguess] = "|00|\n"  else:  T[xguess][yguess] = "|00"  for i in range(len(T)):  for j in range(len(T[i])):  print(T[i][j], end=' ')  print()  print("You hit every ship!") |

## Issues with iteration 1:

1. No input validation
2. No difficulty to select
3. No tutorial or introduction, confusing for new players

## Iteration 2:

|  |
| --- |
| import random  print("Welcome to Battleships!, the game where you try to sink the enemy's ships. You must find and destroy 5 different boats on a 10x10 grid: an Aircraft Carrier which is 5 spaces long, a Frigate which is 4 spaces long, a Cruiser which is 3 spaces long and 2 Destroyers which are 2 spaces long. Developed by 21307114.\nPlease select a difficulty:\n1. Casual\n2. Advanced\n3. Impossible\n")  while True:  difficulty = input("Difficulty: ")  if difficulty.isdigit() and 1 <= int(difficulty) <= 3:  # Valid input  difficulty = int(difficulty)  break  else:  print("Please enter a valid difficulty (1, 2, or 3).")  if difficulty == 1:  lives = 100000  if difficulty ==2:  lives = 50  if difficulty ==3:  lives = 25  #Displays the play area to the user  print(" 0 1 2 3 4 5 6 7 8 9")  T = [  ["0|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["1|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["2|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["3|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["4|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["5|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["6|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["7|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["8|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"],  ["9|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_", "|\_\_|\n"]  ]  #Generates a random co-ord for the Aircraft Carrier to go  UpOrDown = random.randint(0, 1)  if UpOrDown == 0:  aircraftpositionx = random.randint(0, 5)  aircraftpositiony = random.randint(0, 9)  #Maps the Aircraft Carrier to the co-ords given  AircraftTiles = [[aircraftpositionx, aircraftpositiony],  [aircraftpositionx + 1, aircraftpositiony],  [aircraftpositionx + 2, aircraftpositiony],  [aircraftpositionx + 3, aircraftpositiony],  [aircraftpositionx + 4, aircraftpositiony]]  else:  aircraftpositionx = random.randint(0, 9)  aircraftpositiony = random.randint(0, 5)  #Maps the Aircraft Carrier to the co-ords given  AircraftTiles = [[aircraftpositionx, aircraftpositiony],  [aircraftpositionx, aircraftpositiony + 1],  [aircraftpositionx, aircraftpositiony + 2],  [aircraftpositionx, aircraftpositiony + 3],  [aircraftpositionx, aircraftpositiony + 4]]  #Adds all the tiles used to a used tile list  UsedTiles = []  UsedTiles.extend(AircraftTiles)  #Frigate Generation  FrigateTiles = []  #Checks theres no boat in the current position  while not FrigateTiles or any(x in UsedTiles for x in FrigateTiles):  if UpOrDown == 0:  FrigatePositionX = random.randint(0, 6)  FrigatePositionY = random.randint(0, 9)  FrigateTiles = [[FrigatePositionX, FrigatePositionY],  [FrigatePositionX + 1, FrigatePositionY],  [FrigatePositionX + 2, FrigatePositionY],  [FrigatePositionX + 3, FrigatePositionY]]  else:  FrigatePositionX = random.randint(0, 9)  FrigatePositionY = random.randint(0, 6)  FrigateTiles = [[FrigatePositionX, FrigatePositionX],  [FrigatePositionX, FrigatePositionY + 1],  [FrigatePositionX, FrigatePositionY + 2],  [FrigatePositionX, FrigatePositionY + 3]]  UsedTiles.extend(FrigateTiles)  #Cruiser Generation  CruiserTiles = []  while not CruiserTiles or any(x in UsedTiles for x in CruiserTiles):  if UpOrDown == 0:  CruiserPositionX = random.randint(0, 7)  CruiserPositionY = random.randint(0, 9)  CruiserTiles = [[CruiserPositionX, CruiserPositionY],  [CruiserPositionX + 1, CruiserPositionY],  [CruiserPositionX + 2, CruiserPositionY]]  else:  CruiserPositionX = random.randint(0, 9)  CruiserPositionY = random.randint(0, 7)  CruiserTiles = [[CruiserPositionX, CruiserPositionY],  [CruiserPositionX, CruiserPositionY + 1],  [CruiserPositionX, CruiserPositionY + 2]]  UsedTiles = UsedTiles + CruiserTiles  #Destroyer Generation  DestroyerTiles = []  while not DestroyerTiles or any(x in UsedTiles for x in DestroyerTiles):  if UpOrDown == 0:  DestroyerPositionX = random.randint(0, 8)  DestroyerPositionY = random.randint(0, 9)  DestroyerTiles = [[DestroyerPositionX, DestroyerPositionY],  [DestroyerPositionX + 1, DestroyerPositionY]]  else:  DestroyerPositionX = random.randint(0, 9)  DestroyerPositionY = random.randint(0, 8)  DestroyerTiles = [[DestroyerPositionX, DestroyerPositionY],  [DestroyerPositionX, DestroyerPositionY + 1]]  UsedTiles = UsedTiles + DestroyerTiles  #Destroyer Generation  Destroyer2Tiles = []  while not Destroyer2Tiles or any(x in UsedTiles for x in Destroyer2Tiles):  if UpOrDown == 0:  Destroyer2PositionX = random.randint(0, 8)  Destroyer2PositionY = random.randint(0, 9)  Destroyer2Tiles = [[Destroyer2PositionX, Destroyer2PositionY],  [Destroyer2PositionX + 1, Destroyer2PositionY]]  else:  Destroyer2PositionX = random.randint(0, 9)  Destroyer2PositionY = random.randint(0, 8)  Destroyer2Tiles = [[Destroyer2PositionX, Destroyer2PositionY],  [Destroyer2PositionX, Destroyer2PositionY + 1]]  UsedTiles = UsedTiles + Destroyer2Tiles  hits = 0  #Prints the board to the user  for i in range(len(T)):  for j in range(len(T[i])):  print(T[i][j], end=' ')  print()  #Counts how many times a user has hit a ship  while hits < 16:  while True:  xguess = input("Enter row coordinate (0-9): ")  yguess = input("Enter column coordinate (0-9): ")  #Checks to see if the guess made a hit  if xguess.isdigit() and yguess.isdigit():  xguess = int(xguess)  yguess = int(yguess)  if 0 <= xguess <= 9 and 0 <= yguess <= 9:  guessed = [[xguess, yguess]]  break  else:  print("Please enter numbers between 0 and 9.")  else:  print("Please enter valid integer values.")  if any(x in UsedTiles for x in guessed):  print("Hit!")  hits = hits + 1  if yguess == 9:  T[xguess][yguess] = "|XX|\n"  else:  T[xguess][yguess] = "|XX"  else:  print("Miss!")  #Checks the number of lives the player has left  lives = lives-1;  if lives <= 0:  print("You lost!")  break;  if yguess == 9:  T[xguess][yguess] = "|00|\n"  else:  T[xguess][yguess] = "|00"    for i in range(len(T)):  for j in range(len(T[i])):  print(T[i][j], end=' ')  print()  print("You hit every ship!") |

Addressing Iteration 1 issues:

1. Input validation was added and tested:

A screenshot of a computer

Description automatically generated

1. Difficulty Selection was added and validated.
2. Explanation to the game was added to the start of the code.