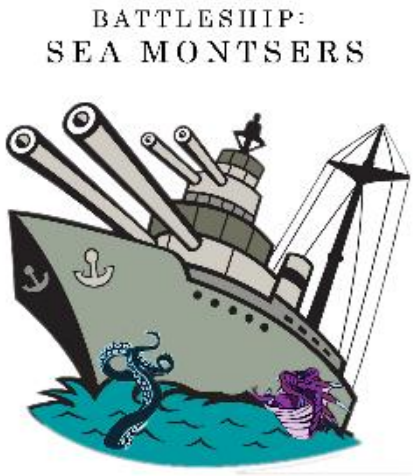


CS1809 Software Design
ASSIGNMENT 2

GAME 4: BATTLESHIP VARIATION B

Usen-ita U. Asanga | 1921943 | Tutor: Mr. Alaa Marshan | Group 12(Yellow)

Contents

Introduction

The assignment brief needed each member of the group to choose a variation out of the four given games. I chose game four – variation b.

the game is titled *Battleship*. The aim of the game is for the player to sink all five ships that are placed on a ten-by-ten square board.

- Aircraft carrier: 5 squares long
- Battleship: 4 squares long
- Submarine: 3 squares long
- Destroyer: 3 squares long
- Patrol Boat: 2 squares long

The tricky part is all ships are invisible. The variation adds two sea monsters, *Kraken* and *Cetus*. This report contains the algorithm of the game design.

Requirements Specification

FUNCTIONAL REQUIREMENTS

The functionalities for the game include.

1. Place the five type of ship of different lengths at random on 10 by 10 board.
2. place Kraken on random unused square
3. place Cetus on random unused square
4. Display empty board, current un-sunk ships and moves.
5. Player clicks the X,Y coordinates of shot.
6. Board updates by adding an 'M' on choses square, if it was a miss, an 'H', if it was a hit, a 'K', if Kaken is hit or a 'C', if Cetus is hit.
7. AI validates shot coordinates then display the appropriate message according to the outcome of the shot:

   - My ship was hit!

   - You missed!

   - You sank my [ship type]!

   - Kraken hit!

   - Cetus hit!

8. AI displays current list of sunk, un-sunk ships and current moves.
9. Player sinks all ships.
10. Display final score.
11. Ask player question "PLAY AGAIN?"
12. Player clicks "QUIT?", go to quite page.
13. AI overlaps two or more ships, redo placement of ships in error.
14. AI place a ship(s) out of bounds, redo placement of ship(s) in error.
15. Click was a miss('M'), take away one from score.
16. Click was a hit('H'), add one to score.
17. Clicks hit('H') or miss('M') square, ignore click.
18. Kraken is hit('K'), score equals zero.
19. Cetus is hit('C'), AI redoes placement of un-suck ships.
20. Sinks Aircraft carrier, add ten to score.
21. Sinks Battleship, add eight to score.
22. Sinks Submarine, add six to score.
23. Sinks Destroyer, add six to score.
24. Sinks patrol Boat, add four to score.

NON-FUNCTIONAL REQUIREMENTS

These are how the game must behave for a good experience.

1. Be appealing and easy to use.
2. Update player's score and moves >0.5 seconds.

Additional Functions Explained

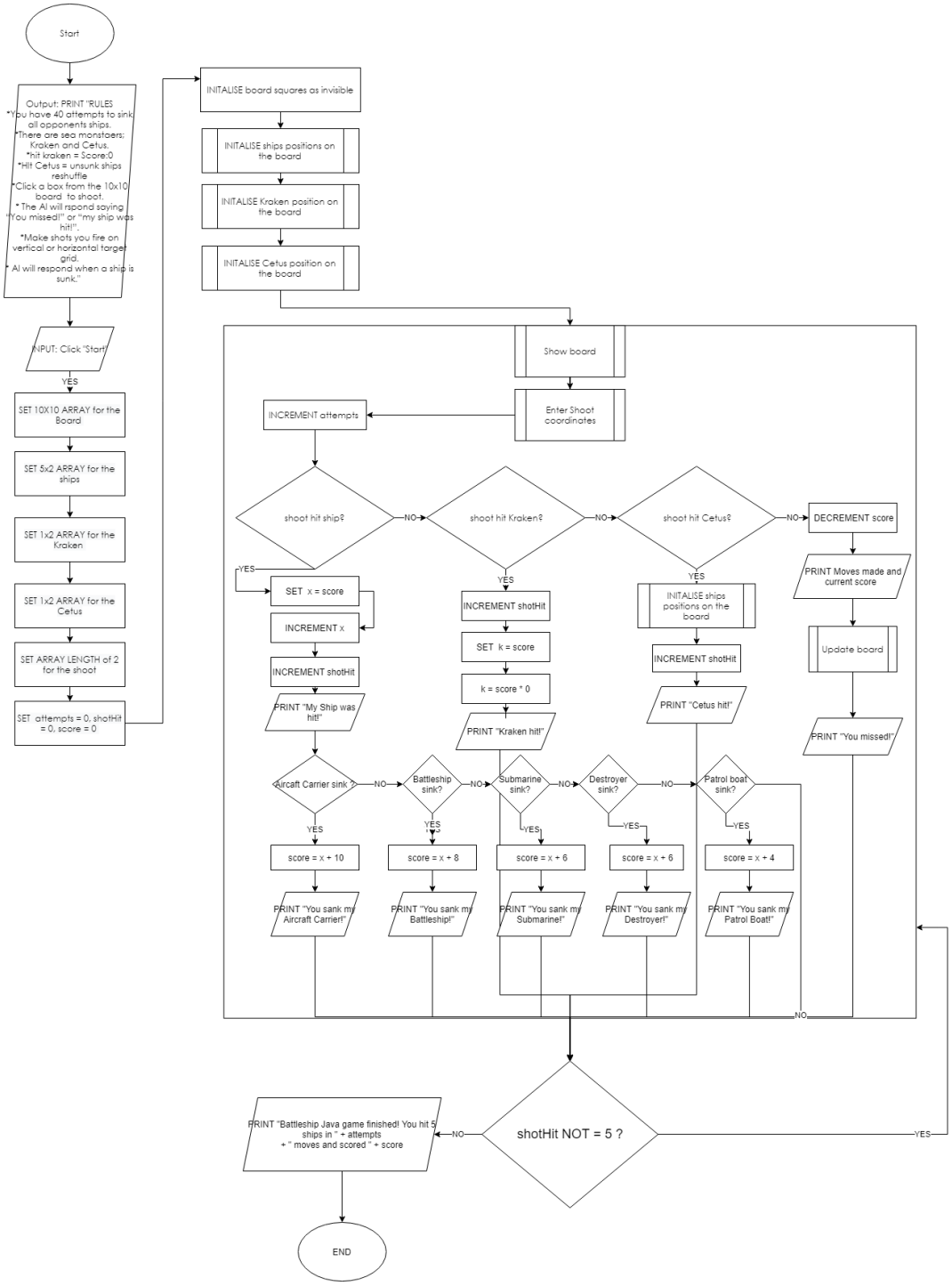
All additional functions above have a dark pink font color.

- Player Moves: At the start, the player number of shots to win the game is counted. Moves increases per shoot. This motivates the player to decipher ways to finish game with less moves.
- Addition of 'K' to hit Kraken and 'C' to hit Cetus: This reduces ambiguity in identifying which shot hit the sea monsters.
- The messages, "Kraken hit!" and "Cetus hit" will let be aware of which monster has been hit.

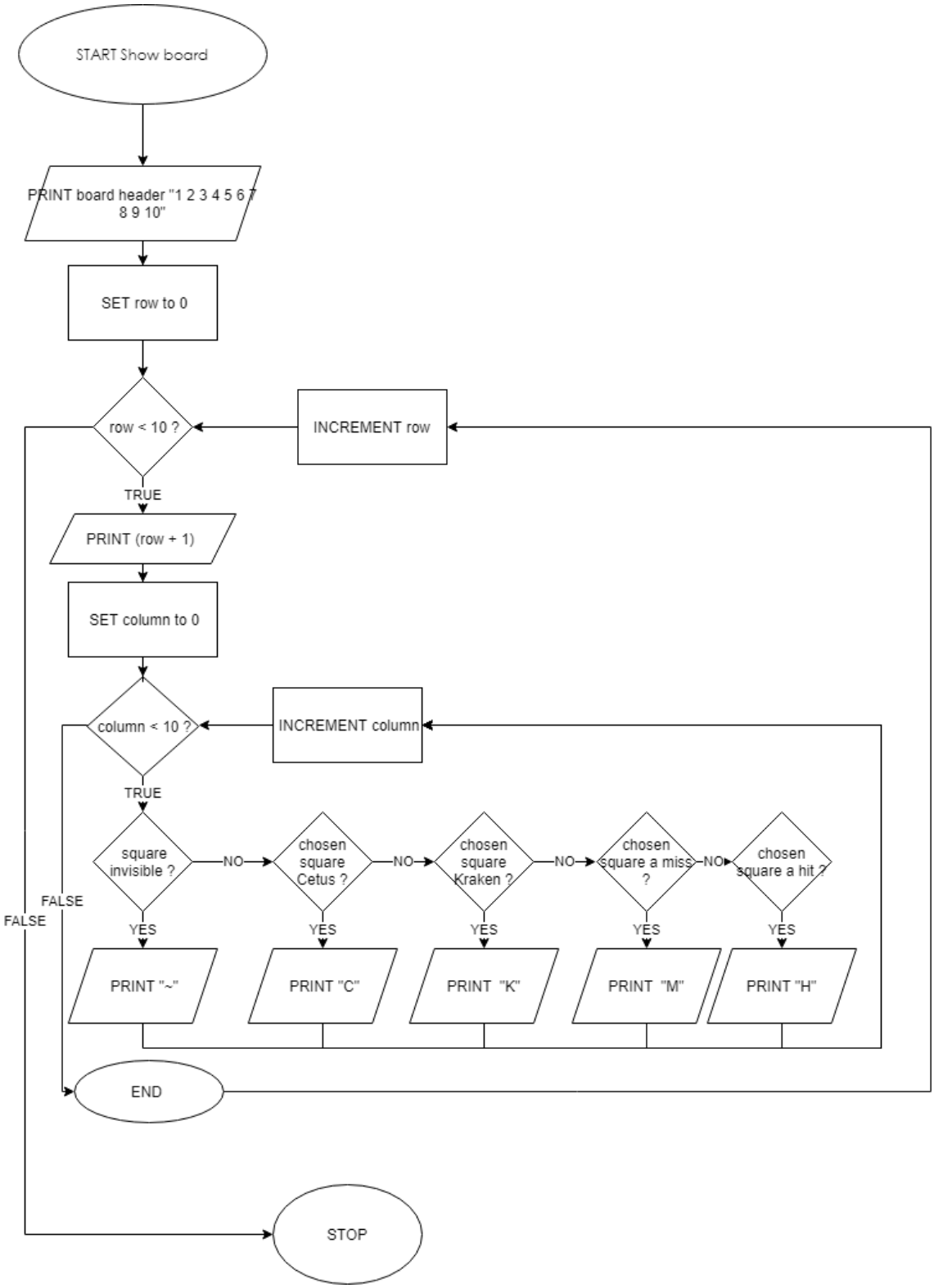
Algorithm Design

FLOWCHART

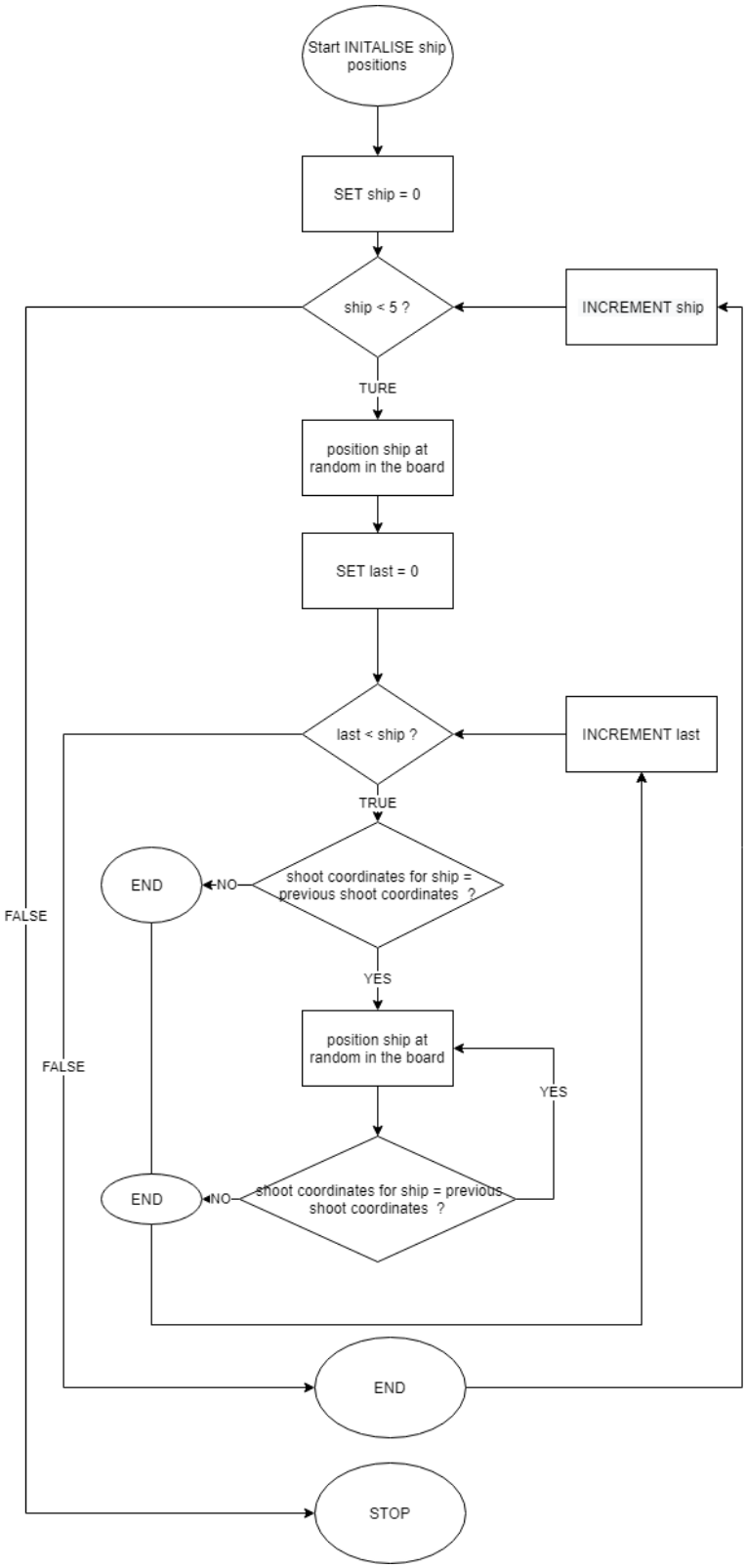
FLOWCHART MAIN:



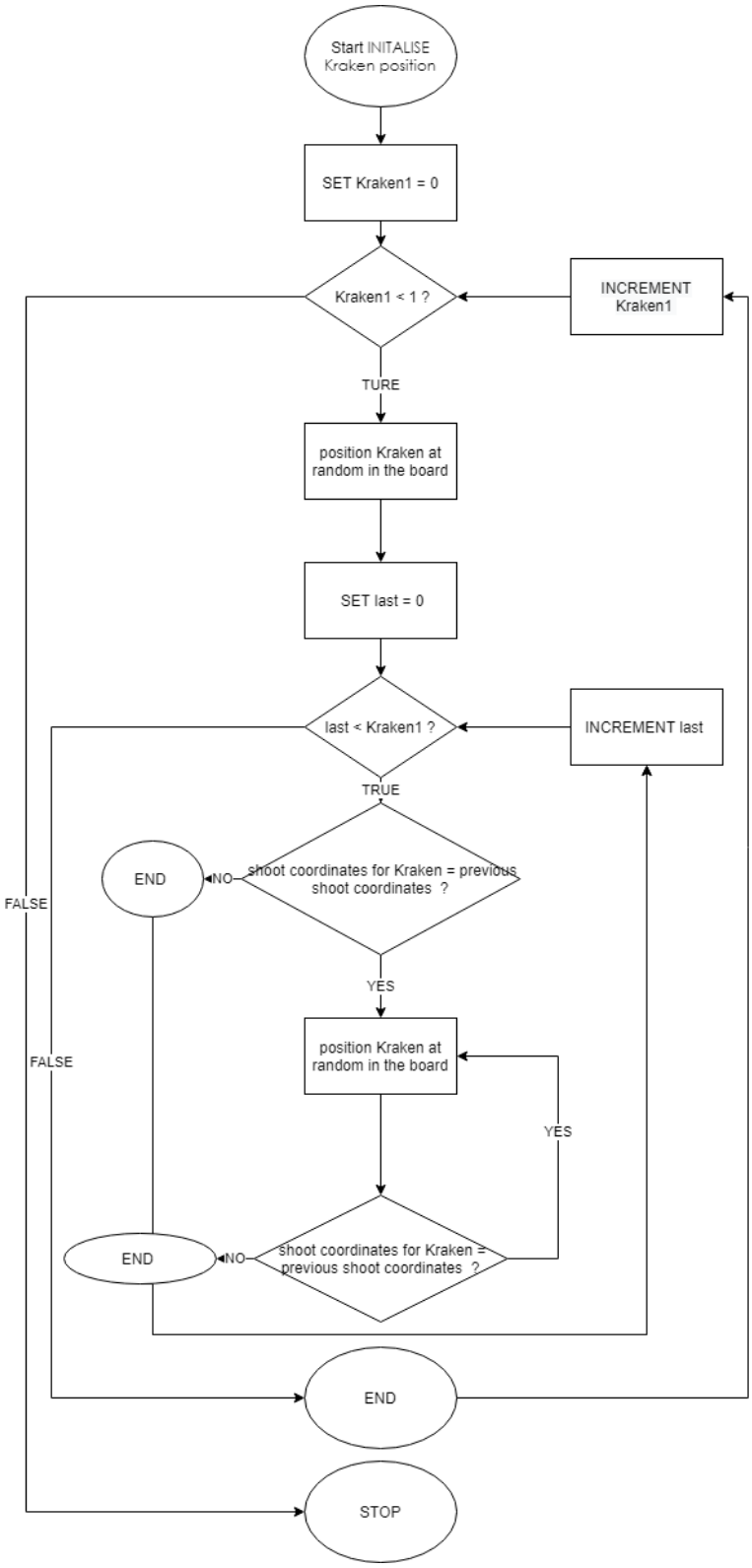
FLOWCHART Show board



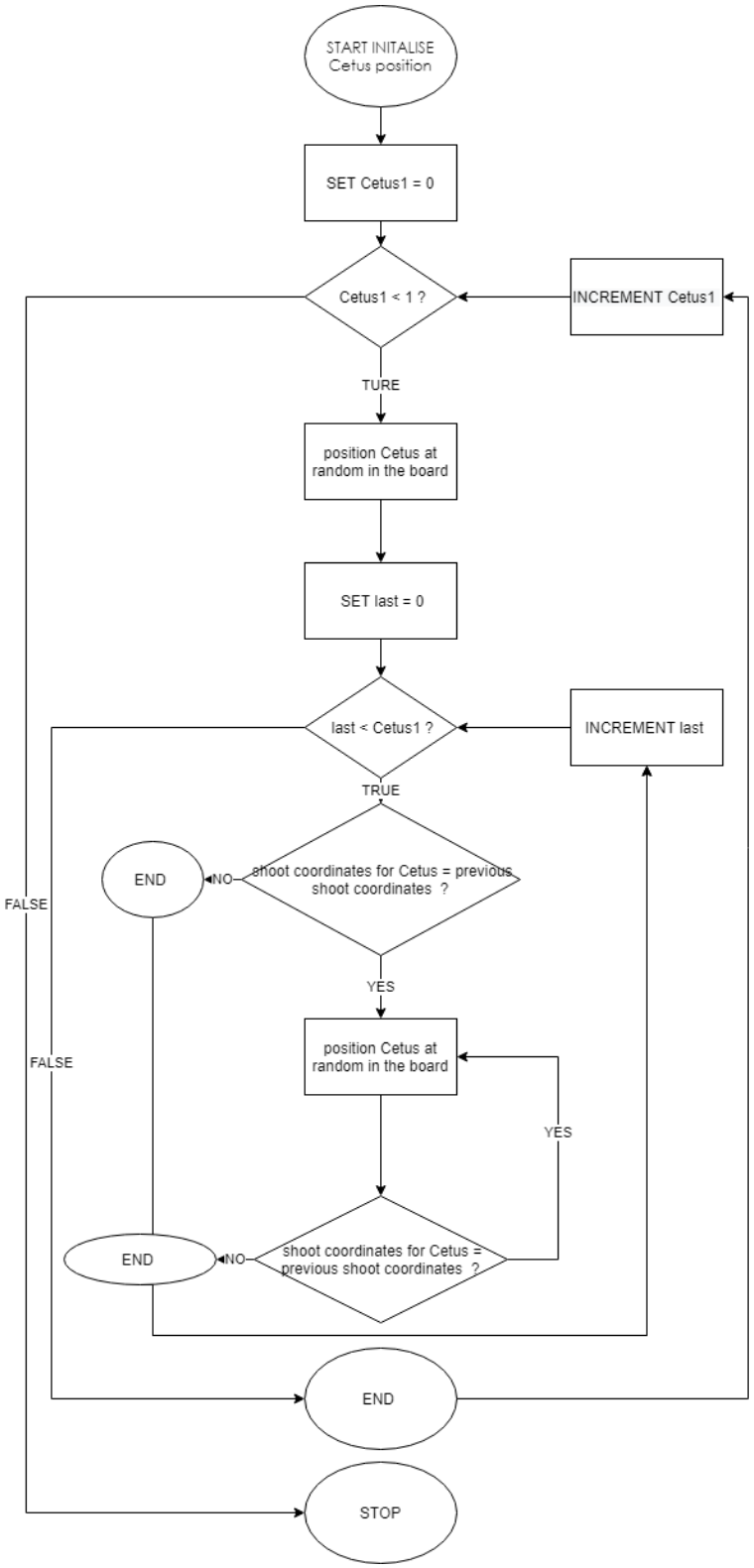
FLOWCHART INITALISE SHIP POSITIONS



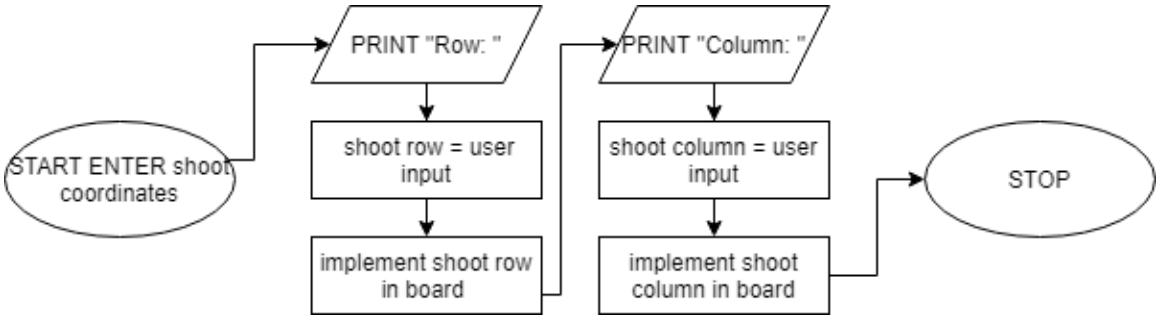
FLOWCHART INITALSIE KRAKEN POSITION



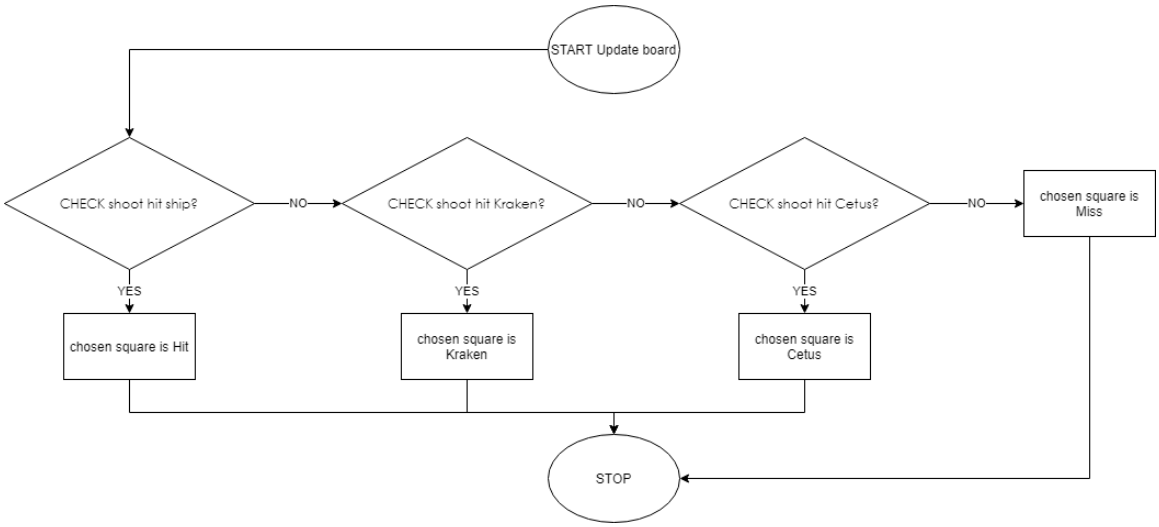
FLOWCHART INITALSIE CETUS POSITION



FLOWCHART ENTER SHOOT COORDINATES



FLOWCHART FUNCTION UPDATE BOARD



PSEUDOCODE

```
Algorithm 1: Rules Screen

START

     Output: PRINT "RULES

             *You have 40 attempts to sink all opponents ships.

             *There are sea monstaers; Kraken and Cetus.

             *hit kraken = Score:0

             *HIt Cetus = unsunk ships reshuffle

             *Click a box from the 10x10 board  to shoot.

             * The AI will rspond saying “You missed!” or “my ship was hit!”.

             *Make shots you fire on vertical or horizontal target grid.

             * AI will respond when a ship is sunk."
```

```
     INPUT "START"
     DISPLAY BattleshipSeaMonsters Screen
END
Algorithm 2: BattleshipSeaMonsters Screen
START
    COMPUTE:
       1)SET 10X10 ARRAY for the Board
       2)SET 5x2 ARRAY for the ships
       3)SET 1x2 ARRAY for the Kraken
       4)SET 1x2 ARRAY for the Cetus
       5)SET ARRAY LENGTH of 2 for the shoot
       6)SET  attempts = 0, shotHit = 0, score = 0
       7)FUNCTION INITALISE EMPTY board:
                  SET 10X10 board
                  FOR row IN 0 TO 10
                   FOR column IN 1 TO 10
                    DO Board row & column = -3
                   ENDFOR;
                  ENDFOR;
         END FUNCTION
       8)FUNCTION INITALISE ships positions:
                  FOR ship IN 0 TO 5
                   DO position ship row at random between 1 & 10
                      position ship row at random between 1 & 10
                    FOR last IN 0 TO ship
                     IF shoot coordinates for ship row and column = previous shoot
                        coordinates for ship
                         THEN DO
                               position ship row at random between 1 & 10
```

```
                        position ship row at random between 1 & 10
                       WHILE (shoot coordinates for ship row and column =
                              previous shoot coordinates for ship)
              END IF
             ENDFOR;
            ENDFOR;
  END FUNCTION
9)FUNCTION INITALISE Kraken positions:
           FOR Kraken1 IN 0 TO 1
            DO position Kraken row at random between 1 & 10
               position Kraken row at random between 1 & 10
             FOR last IN 0 TO Kraken1
              IF shoot coordinates for Kraken row and column = previous shoot
                 coordinates for Kraken
                  THEN DO
                        position Kraken row at random between 1 & 10
                        position Kraken row at random between 1 & 10
                       WHILE (shoot coordinates for Kraken row and column =
                              previous shoot coordinates for Kraken)
              END IF
             ENDFOR;
            ENDFOR;
  END FUNCTION
10)FUNCTION INITALISE Cetus positions:
           FOR Cetus1 IN 0 TO 1
            DO position Cetus row at random between 1 & 10
               position Cetus row at random between 1 & 10
             FOR last IN 0 TO Cetus1
              DO IF shoot coordinates for Cetus row and column = previous shoot
```

```
                    coordinates for Cetus
                  THEN DO
                        position Cetus row at random between 1 & 10
                        position Cetus row at random between 1 & 10
                       WHILE (shoot coordinates for Cetus row and column =
                              previous shoot coordinates for Cetus)
                 END IF
             ENDFOR;
            ENDFOR;
  END FUNCTION
11)DO
    FUNCTION Show board:
           OUTPUT PRINT "1 2 3 4 5 6 7 8 9 10"
           FOR row IN 0 TO 10
            DO PRINT(row + 1) + " "
             FOR column IN 0 TO 10
              DO IF Board row & column = -3
                   THEN PRINT " " + "~"
                  ELSE IF Board row & column = -2
                        THEN PRINT " " + "C"
                  ELSE IF Board row & column = -1
                        THEN PRINT " " + "K"
                  ELSE IF Board row & column = 0
                        THEN PRINT " " + "M"
                  ELSE IF Board row & column = 1
                        THEN PRINT " " + "H"
                  END IF
             END FOR
           END FOR
```

```
END FUNCTION
FUNCTION Enter Shoot coordinates:
        OUTPUT PRINT "Row: "
        COMPUTE: shoot row = user input
                 shoot row DECREMENT
        OUTPUT PRINT "Column: "
        COMPUTE: shoot column = user input
                 shoot column DECREMENT
END FUNCTION
attempts INCREMENT
IF shoot hit ship
 THEN SET x=score
      x INCREMENT
      shotHit INCREMENT
      OUTPUT PRINT "My Ship was hit!"
      IF ships.length = 1
       THEN score = x + 10
      ELSE IF ships.length = 2
            THEN score = x + 8
      ELSE IF ships.length = 3
            THEN score = x + 6
      ELSE IF ships.length = 4
            THEN score = x + 6
      ELSE IF ships.length = 5
            THEN score = x + 4
      END IF
      FUNCTION Print "you sank" + ship type:
               FOR ship IN 0 TO ships.length
                DO IF User input = ship position & ship = 0
```

```
                    THEN PRINT "You sank my Aircraft Carrier!"
                    ELSE IF User input = ship position & ship = 1
                          THEN PRINT "You sank my Battleship!"
                    ELSE IF User input = ship position & ship = 2
                          THEN PRINT "You sank my Submarine!"
                    ELSE IF User input = ship position & ship = 3
                          THEN PRINT "You sank my Destroyer!"
                    ELSE IF User input = ship position & ship = 4
                          THEN PRINT "You sank my Patrol Boat!"
                    END IF
               END FOR;
      END FUNCTION
ELSE IF shoot hit Karken
      THEN shotHit INCREMENT
           SET k = score
           k = score * 0
           OUTPUT PRINT "Kraken hit!"
ELSE IF shoot hit Cetus
      THEN FUNCTION INITALISE ships positions:
            FOR ship IN 0 TO 5
             DO position ship row at random between 1 & 10
                position ship row at random between 1 & 10
              FOR last IN 0 TO ship
               IF shoot coordinates for ship row and column = previous shoot
                  coordinates for ship
                THEN DO
                     position ship row at random between 1 & 10
                     position ship row at random between 1 & 10
                     WHILE (shoot coordinates for ship row and column =
```

```
                                    previous shoot coordinates for ship)
                      END IF;
                     ENDFOR;
                   ENDFOR;
                  END FUNCTION
                  shotHit INCREMENT
                  OUTPUT PRINT "Cetus hit!"
     ELSE score DECREMENT
          FUNCTION Print current attempts& score:
                  OUTPUT PRINT "Moves: " + attempt + "Score: " + score
          END FUNCTION
          FUNCTION Update board:
                  IF shoot hit ship
                   THEN COMPUTE: Board row & column = 1
                  ELSE IF shoot hit Kraken
                        THEN COMPUTE: Board row & column = -1
                  ELSE IF shoot hit Cetus
                        THEN COMPUTE: Board row & column = -2
                  ELSE COMPUTE: Board row & column = 0
                  END IF;
          END FUNCTION
          PRINT "You missed!"
     END IF
12)WHILE(shotHit != 5)
13)FUNCTION Show board:
           OUTPUT PRINT "1 2 3 4 5 6 7 8 9 10"
           FOR row IN 0 TO 10
            DO PRINT(row + 1) + " "
             FOR column IN 0 TO 10
```

```
                    DO IF Board row & column = -3
                        THEN PRINT " " + "~"
                       ELSE IF Board row & column = -2
                             THEN PRINT " " + "C"
                       ELSE IF Board row & column = -1
                             THEN PRINT " " + "K"
                       ELSE IF Board row & column = 0
                             THEN PRINT " " + "M"
                       ELSE IF Board row & column = 1
                             THEN PRINT " " + "H"
                       END IF
                   END FOR
                 END FOR
         END FUNCTION
      14)OUTPUT PRINT ("Battleship Java game finished! You hit 5 ships in " + attempts
                        + " moves and scored " + score)
END
Algorithm 3: GameOver Screen
START
     DISPLAY INTEGER FinalScore from BattleshipSeaMonsters Screen, INTEGER score
      INPUT "PLAY AGAIN?"
        OUTPUT Rules Screen
END
```