

# Battleship

DESHAUN ORMOND

# Introduction

- ▶ Two player guessing game.
- ▶ 10 x 10 grid
- ▶ The location of the fleet are concealed from the other player
- ▶ Player alternate turns until one destroys the opposing fleet.
- ▶ 5 ships
  - ▶ Carrier = 5
  - ▶ Battleship = 4
  - ▶ Cruiser = 3
  - ▶ Submarine = 3
  - ▶ Destroyer = 2

# Making the Board

```
def get_user_input():
    while(True):
        try:
            #x,y =[int(i) for i in input("Please enter coordinates(x,y): ")]
            x = int(input("Please select a row:"))
            y = int(input("Please select a column:"))
            x = x-1;
            y = y-1;

            if (x >9 or y >9 or x < 0 or y < 0):
                print "Please enter numbers from 1-10"
            else:
                return x,y
        except NameError:
            print "You did not enter a number, try again"
        except TypeError :
            print "Something went wrong,let's try again"
        except SyntaxError:
            print "Something went wrong,let's try again"
        |

def get_ori():
    while(True):
        ori = raw_input("Would you like your ship vertical or horizonatal, (v,h) ").lower()
        if ori == "v" or ori == "h":
            print ori;
            return ori

        else:
            print "Wrong letter, try again"
```

## User Input and Orientation

```

def place_battleship():

    while (True):
        print("Let's place your Battleship")
        i = 2;
        ship = "B";
        print("Your ship length will be 4")
        origlength = 3; #numbers in computer sciece start at 0
        user_row,user_col = get_user_input()
        user_orientation = get_ori()
        if (oriation_check(user_orientation,user_row,user_col,origlength,ship,i) == True):
            break;
        else:
            print"Uh oh, That coordinate overlaps other ships or does not fit on the board\nLet's try again! "
            print_user_board(user_board)

```

## Placing ships

```
def oriation_check(user_orientation,user_row,user_col,origlength,ship,i):
    if i == 1:#Battleship
        if (user_row < 0 or user_row >= 7 or user_col < 0 or user_col >= 7 ):
            print "You are placing the Battleship\nSo your ship size will be four"
            print "The spot you picked will not fit on your board"
            print "Try and enter some numbers between 1-7 "
            print_user_board(user_board)
            return False;
        else:
            if (ship_placement(user_row,user_col,user_orientation,origlength,ship,i) == True):
                return True;
```

## Orientation Check



```

def ship_placement(user_row,user_col,user_orientation,origlength,ship,i):
    count = 0;
    if (user_orientation == "v" ):
        while (origlength >= 0):
            if (user_board[user_row][user_col] != "0" ):##if there is NOT an open spot or it overlaps
                while( count > 0): ##Erase whatever letter was put on the board earlier
                    user_board[user_row-1][user_col] = "0"
                    user_row = user_row -1;
                    count = count -1;
                return False;

            else: # If open spot then continue
                user_board[user_row][user_col] = ship; ##The letter previously assigned in parent fucntion
                user_row = user_row +1;
                count = count +1
                origlength = origlength - 1;
        return True;

```

# User ship placement

```

def computer_place_ships(computer_board, ships):

    for i,j in ships.items():
        ship_not_place = 0
        x = random.randint(0, len(computer_board)-1)
        y = random.randint(0, len(computer_board)-1)
        ori = 0
        #random.randint(0,1)
        while(ship_not_place < len(computer_board)-1):
            try:
                if ori == 0:
                    print "placing", i,x,y
                    if computer_board[x][y] == "O":

                        for k in range(j):
                            computer_board[x+k][y] = i
                            ship_not_place = ship_not_place + 1
                        print "placed", i
                        break;
                    elif computer_board[x][y] != "O":
                        #computer_board[x][y] = "O"
                        for k in range(j):
                            computer_board[x-k][y] = "O"
                        break;

            except IndexError:
                computer_board[x][y] = "O"
                print "Error", x+1,y+1
                if ori == 0:
                    #computer_board[x][y] = "O"
                    for k in range(j):
                        computer_board[x-k][y] = "O"

                break;

```

# Computer placing ships



```
def user_guess(computer_board, comp_x, comp_y):
    hits = 0
    print_computer_board(computer_board)

    while(True):
        if hits < 17:
            user_x, user_y = get_user_input()
            if ( user_x == comp_x and user_y == comp_y):
                computer_board[user_x][user_y] = "X"

                print_computer_board(computer_board)
                print "You hit your opponenents ship!"
                print "Guess again!"
                raw_input("Press Enter to continue...")
                hits = hits +1;

            else:
                computer_board[user_x][user_y] = "M"
                print_computer_board(computer_board)
                print "You missed your opponent's ship"
                print "Now it's your opponent's turn"
                raw_input("Press Enter to continue...")
                break;

        else:
            print "Congratulations! You sunk all of your opponenents ships "
            print "YOU WIN"
            print_computer_board(computer_board)
            sys.exit(0)
```

## User Guessing

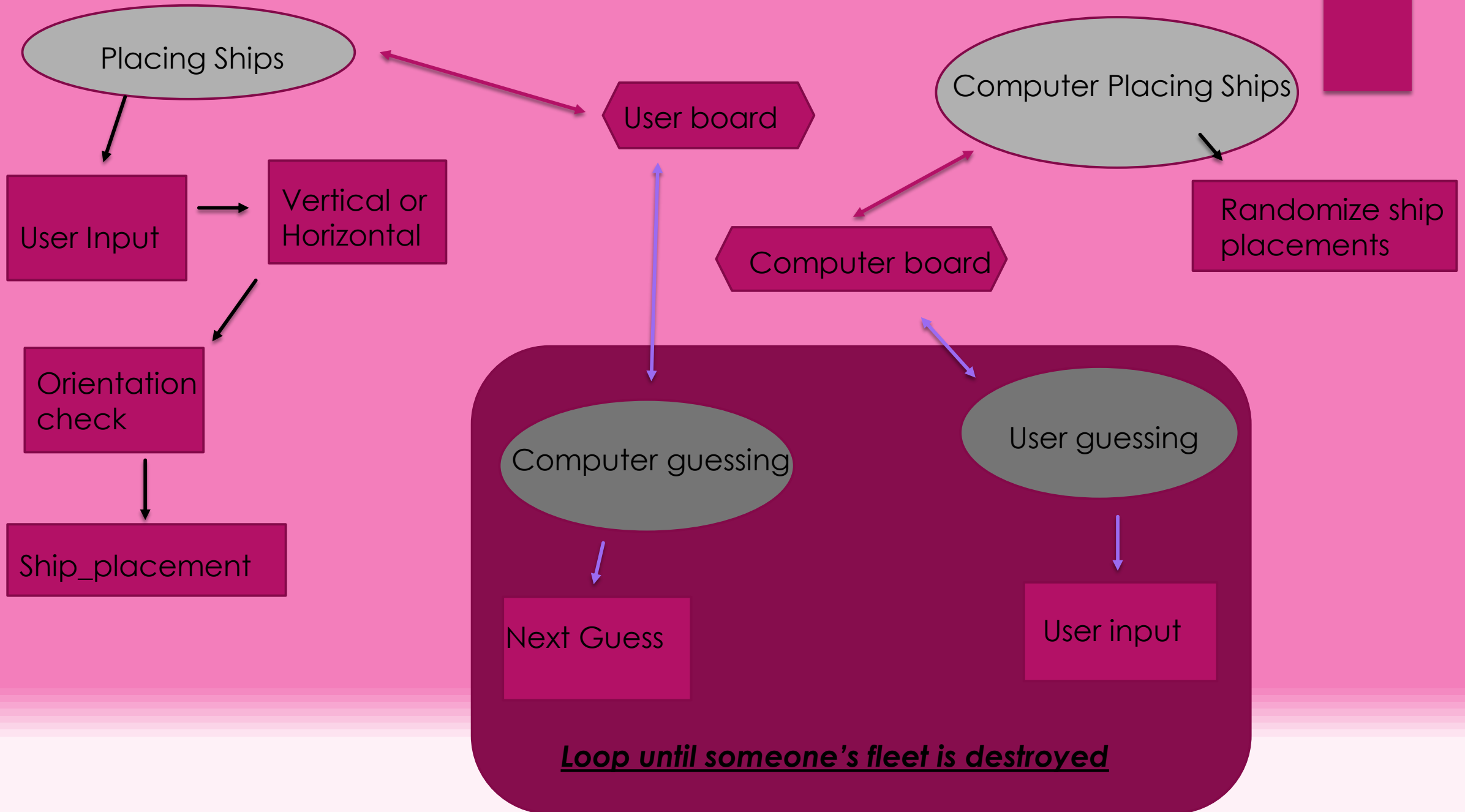
```
def comp_guess(user_board):
    hits = 0;
    while (True):
        if hits < 17:
            print "The computer is guessing....."
            comp_guess_x = random.randint(0,9);
            comp_guess_y = random.randint(0,9);
            time.sleep(1)
            if (user_board[comp_guess_x][comp_guess_y] != "O" ):
                print "The computer hit your ship!! "
                user_board[comp_guess_x][comp_guess_y] = "X"
                print_user_board(user_board)
                print "The computer will guess again"
                raw_input("Press Enter to continue...")
                hits = hits + 1;
                nextguess(user_board,comp_guess_x,comp_guess_y)
            else:
                print "The computer missed your ship"
                user_board[comp_guess_x][comp_guess_y] = "M"
                print_user_board(user_board)
                print "Now it is your turn"
                raw_input("Press Enter to continue...")
                break;
        else:
            print "Your opponent hit all of your ships "
            print "YOU LOSE"
            print_user_board(user_board)
            sys.exit(0)
```

# Computer Guessing

```
def nextguess(user_board, comp_guess_x, comp_guess_y, hits):  
    smartguess = -1;  
  
    while (smartguess < 0):  
  
        if (user_board[comp_guess_x]<9 and user_board[comp_guess_x][comp_guess_y]!="O"): ## x+1,y  
            hits = hit +1;  
            user_board[comp_guess_x+1][comp_guess_y] = "X"  
            print_user_board(user_board)  
            print "The computer hit your ship!!"  
            print "The computer guessed" , comp_guess_x+1 , "," , comp_guess_y  
            print "The computer will guess again"  
            raw_input("Press Enter to continue...")  
  
        else:  
  
            user_board[comp_guess_x][comp_guess_y] = "M"  
            print_user_board(user_board)  
            print "The computer missed your ship"  
            print "Now it is your turn"  
            smartguess = 1;  
            raw_input("Press Enter to continue...")  
            break;
```

## Smart Guess

## Battleship Flow Chart





# Display



# References

- ▶ <http://effbot.org/zone/stupid-exceptions-keyboardinterrupt.htm>
- ▶ <https://docs.python.org/3.1/tutorial/datastructures.html>
- ▶ <http://code.activestate.com/recipes/578836-the-game-of-battleships-in-python/>
- ▶ <https://www.codecademy.com/courses/learn-python/lessons/python-lists-and-dictionaries/exercises/access-by-index-1>
- ▶ <https://stackoverflow.com/questions/27310631/checking-if-input-is-an-integer>