

TITLE:

Brendan Uebelhoer
 CSCI 101 Section A
 Pegboard simulator

PROGRAM:

I intend to write a program that simulates playing the classic pegboard game, in which a board with a grid of hole is filled with pegs, leaving only one hoe open. The game is played by taking a peg and “jumping” another peg into the empty hole. The jumped peg is then removed, leaving two open holes. The game is then repeated until there are no more possible moves. The game is scored by the number of remaining pegs, with 1 being the highest score. I intend to implement all the formal rules of the game, with the ability to select a premade board or import your own. Other possible features include a timing system, scoreboard, and different difficulties that can suggest moves for you

Collection:

```
def getboard(file):      #imports a board to a nested list from a .csv file
    with open(file, 'r') as csvfile:
        board = []
        i=0
        boardreader = csv.reader(csvfile)
        for row in boardreader:
            board.append([])
            board[i] = row
            j = 0
            for spot in board[i]:
                board[i][j] = int(spot)
                j += 1
            i += 1
    return board
```

This function takes a filename and imports the cvs file of that name and returns it as a 2D list that the rest of the program can interact with and change, with the first list serving as the Y coordinates of the list and the second list as X coordinates.

Algorithm:

```
def printboard(printedboard):      #prints out the parameterized board
    r=0
    print(' ', end = ' ')
    for i in range(len(printedboard[0])):
        print(i, end = ' ')
    print()
    for i in printedboard:
        print(r, end = ' ')
        r += 1
        for j in i:
```

```
    if j == -1:
        print(' ', end=' ')
    if j == 0:
        print('o', end=' ')
    if j == 1:
        print('i', end=' ')
    print()
```

This function prints the board that it is inputted, translating the numerical markers used internally to symbols and letters that are easier for a user's to visually see. This is important because printing a board is a very common task in my program, and it is a fairly complicated process, so making it a function vastly simplifies my code. It takes each element of each list, and translates it from a number (-1, 0, or 1) into a letter or a blank that visually looks better. It also prints numbers alone the sides to help the used easily identify coordinates of the grid. The function also scales theoretically infinitely. If the element returned by the double "for" loops is a "-1", the function will print a space, followed by another space. If the value is a "1" it will print a "i", followed by a space, and if the input is "0" it will print a "o" followed by a space. This function does not check the validity of an input, as nothing inputted by this function would come from a user, all values would be defined in the code somewhere else, and so unexpected values are not likely.