

Overview

The objective of this lab was to explore some uses of stacks and queues. It's important to understand when certain data structures should be used because the efficiency of a program can be significantly affected by what data structure is used. We coded a game of "Towers of Hanoi." This game made use of the Stack class because the towers are literally stacks of items. Items must always be added and removed from the top.

Task 2

```

λ a
1 = 1
1 = 1
1 = 1
Welcome to the Tower of Hanoi
There are 3 basic rules:
1) Only one disk may be moved at a time.
2) Each move must consist of taking the top disk from a stack and moving it to another stack.
3) No larger disk may be placed onto a smaller disk.
Let's begin!

How many disks are on the stack? 3
<== Tower 1 ==>
1
2
3
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
-----

Which stack would you like to take from? 1
Which stack would you like to move to? 2
<== Tower 1 ==>
2
3
-----
<== Tower 2 ==>
1
-----
<== Tower 3 ==>
-----

Which stack would you like to take from? 2
Which stack would you like to move to? 3
<== Tower 1 ==>
2
3
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
1
-----

```

Complete this before moving on to task

Task 2: Create a single person version

1. Read the introduction to the game
2. Create a main program that uses
 - a. It needs to determine the
 - b. It needs to provide a way
 - c. It needs to prevent illegal
 - d. It needs to provide the u
3. Test your program to ensure it meets success conditions and expected
4. Include in the lab report a screen

Task 3: Make a game of it.

1. Create a new project
2. Copy any code from the previous
3. In this version, there are 3 towers

Which stack would you like to take from? 1
Which stack would you like to move to? 2
<== Tower 1 ==>
3

<== Tower 2 ==>
2

<== Tower 3 ==>
1

Which stack would you like to take from? 3
Which stack would you like to move to? 2
<== Tower 1 ==>
3

<== Tower 2 ==>
1

<== Tower 3 ==>
2

Which stack would you like to take from? 1
Which stack would you like to move to? 3
<== Tower 1 ==>

<== Tower 2 ==>
1

<== Tower 3 ==>
2

Which stack would you like to take from? 2
Which stack would you like to move to? 1
<== Tower 1 ==>

<== Tower 2 ==>
1

<== Tower 3 ==>
2

<== Tower 3 ==>
3

Sharepoint.com

Complete th

Task 2: Cr

1. Read

2. Crea

3. Test

4. Incl

5. Test

6. Incl

7. Test

8. Incl

9. Test

10. Incl

Task 3: Ma

1. Crea

2. Copy

3. In th

4. Test

5. Incl

6. Test

7. Incl

8. Test

9. Incl

10. Test

11. Incl

12. Test

13. Incl

14. Test

15. Incl

16. Test

17. Incl

18. Test

19. Incl

20. Test

21. Incl

22. Test

23. Incl

24. Test

25. Incl

26. Test

27. Incl

28. Test

29. Incl

30. Test

31. Incl

32. Test

33. Incl

34. Test

35. Incl

```

Which stack would you like to take from? 2
Which stack would you like to move to? 1
<== Tower 1 ==>
1
-----
<== Tower 2 ==>
2
-----
<== Tower 3 ==>
3
-----

Which stack would you like to take from? 2
Which stack would you like to move to? 3
<== Tower 1 ==>
1
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
2
3
-----

Which stack would you like to take from? 1
Which stack would you like to move to? 3
<== Tower 1 ==>
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
1
2
3
-----

```

Task 3

When there are two players, it works best to cooperate. Otherwise, It's very difficult to move anything anywhere. It's goes the fastest when one player stays out of the way of the other player while he moves to the other side. Then it is clear for the last player to complete his stack.

```

Welcome to the Tower of Hanoi
There are 3 basic rules:
1) Only one disk may be moved at a time.
2) Each move must consist of taking the top disk from a stack and moving it to another stack.
3) No larger disk may be placed onto a smaller disk.
Let's begin!

How many disks are on the stack? 3
<== Tower 1 ==>
1 -> Player 1
2 -> Player 1
3 -> Player 1
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
-----
<== Tower 4 ==>
-----
<== Tower 5 ==>
1 -> Player 2
2 -> Player 2
3 -> Player 2
-----
It's Player 1's turn

Which stack would you like to take from? 1
Which stack would you like to move to? 3
<== Tower 1 ==>
2 -> Player 1
3 -> Player 1
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
1 -> Player 1
-----
<== Tower 4 ==>
-----
<== Tower 5 ==>
1 -> Player 2
2 -> Player 2
3 -> Player 2
-----
It's Player 2's turn

```

```

Which stack would you like to take from? 2
Which stack would you like to move to? 4
<== Tower 1 ==>
1 -> Player 2
2 -> Player 2
3 -> Player 2
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
1 -> Player 1
-----
<== Tower 4 ==>
2 -> Player 1
3 -> Player 1
-----
<== Tower 5 ==>
-----
It's Player 2's turn

Which stack would you like to take from? 3
Which stack would you like to move to? 4
<== Tower 1 ==>
1 -> Player 2
2 -> Player 2
3 -> Player 2
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
-----
<== Tower 4 ==>
1 -> Player 1
2 -> Player 1
3 -> Player 1
-----
<== Tower 5 ==>
-----
Good job. You moved the towers!

```

Task 4

We created a structure to represent a move. It has the location the block was moved from and the location it was moved to. Every time we moved a block, we recorded this move and added it to the queue. At the end, we emptied the queue and printed the moves in the same order they were performed. We implemented an automatic solution using the link. We had to change the characters to integers to use them as indices for our array. We recorded these moves into a queue and printed them to the screen.

```
How many disks are on the stack?
Here's the automated solution
<== Tower 1 ==>
1
2
3
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
-----
Next Move: 1 -> 2
<== Tower 1 ==>
2
3
-----
<== Tower 2 ==>
1
-----
<== Tower 3 ==>
-----
Next Move: 1 -> 3
<== Tower 1 ==>
3
-----
<== Tower 2 ==>
1
-----
<== Tower 3 ==>
2
-----
Next Move: 2 -> 3
<== Tower 1 ==>
3
-----
<== Tower 2 ==>
-----
<== Tower 3 ==>
1
2
-----
Next Move: 1 -> 2
```

```
Next Move: 3 -> 1
<== Tower 1 ==>
1
-----
<== Tower 2 ==>
3
-----
<== Tower 3 ==>
2
-----
Next Move: 3 -> 2
<== Tower 1 ==>
1
-----
<== Tower 2 ==>
2
3
-----
<== Tower 3 ==>
-----
Next Move: 1 -> 2
<== Tower 1 ==>
-----
<== Tower 2 ==>
1
2
3
-----
<== Tower 3 ==>
-----
here's a summary of the moves
1 -> 2
1 -> 3
2 -> 3
1 -> 2
3 -> 1
3 -> 2
1 -> 2
Your turn!
```