# Game Technology – Tower of Hanoi (Resit)

**Student name:** Lionel Mithoe

**Student number:** 500691511

**Course:** BU Project Game Technology

**Assignment:** Tower of Hanoi in C++

**Date:** March 15th 2016

## My solution

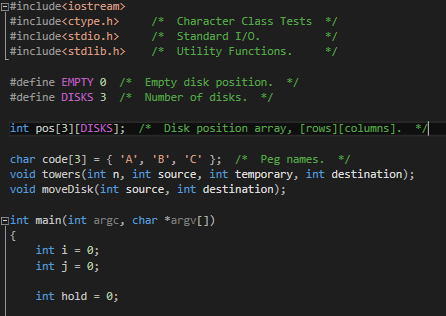
To solve the Tower of Hanoi problem for this assignment I decided to look up a few articles about how to solve the Tower of Hanoi in a non-programmable way. The general way of solving the problem is by dividing the stack on the far left over to the pegs in the middle and the far right. The peg on the far left should eventually have only the largest disc left which you will try to move over to the peg on the far right by stacking all the other discs on the middle stack. Then you solve the Tower of Hanoi by stacking the rest of the discs onto the largest disc which is now located on the peg on the far right.

After being able to solve this problem without a visual representation I tried drawing a grid in ASCII, but this didn’t work out for me at first. After having looked over a few tutorials on how to do this I implemented some of the code and adjusted it to something that makes more sense to me.

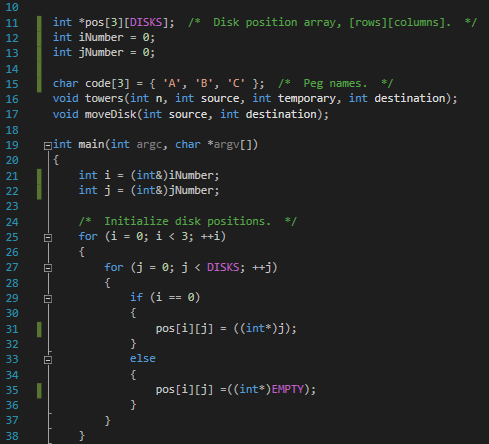
## Requirements

A brief explanation of how I met the requirements for this assignment:

**Programming language has to be C++**Shown in the image below, all the programming for this assignment was done in C++

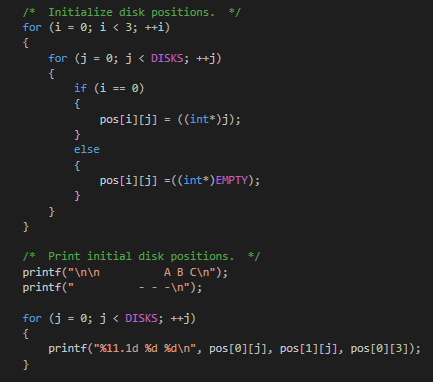


*Using an important function of C++ called “main” and various “includes” this program was able to function correctly.*

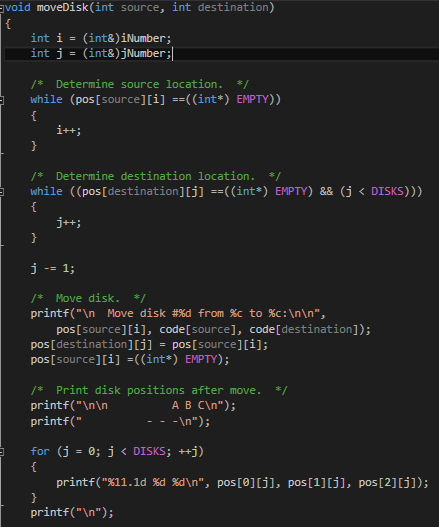
**The program makes use of pointers and references**The array called pos is a pointer and points to its values during the for loop which creates the grid’s rows and columns. The int called i and the int called j refer to the int called iNumber and the int called jNumber respectively which decides where on the grid the disks are placed. 

**There is a visual representation**

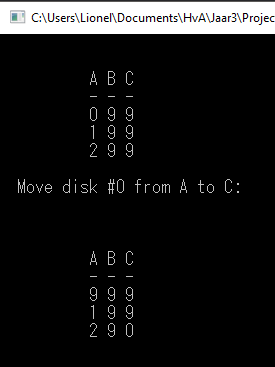
This piece of code prints out the first display of the Tower where all of the disks are stacked in the correct order at the first peg. I choose to put this separately because I wasn’t moving the disks yet at this point.



This is where the moving disks get printed. First I locate where the disks are stored at the moment, then I look at where the disk should be placed and after that the result of that gets printed alongside a short message telling you what happened



This is what it looks like visually



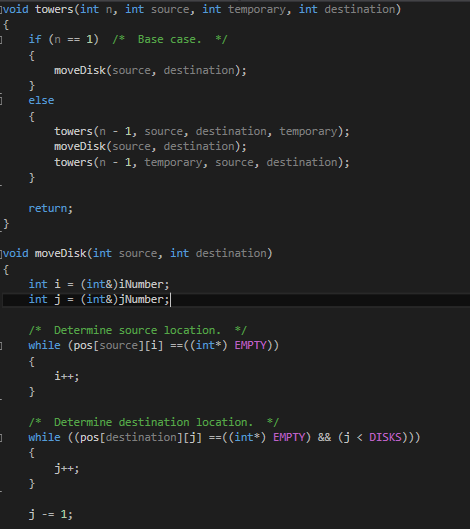
**There are at least 3 disks**

The most basic commonly used form of the Tower of Hanoi game uses 3 disks, therefore I opted to start out with 3 disks and was planning on expanding it to at least 8 by letting the user input how many disks they’d want to use. I however didn’t have enough time to work on this so I stuck to 3. The 9’s represent empty spaces and the 0, 1, 2 represent the disks from smallest to biggest, respectively.



**The disc movement follows the three rules**

The tower follows the basic rule of how to solve the Tower of Hanoi problem. By stacking the disks from the source peg to the temporary peg and destination peg until the source peg is left with only the largest disk, constantly checking the values of each disk before each move to make sure that a disk with a higher value isn’t placed atop of a disk with a lower value. The disks then stack themselves unto the temporary peg in order of highest to lowest value to make sure that the destination peg is empty for the highest valued disk which is still residing on the source peg can be placed there. Then the remaining disks bounce between the source and temporary pegs so they can be place on the destination peg in order from highest to lowest value, solving the Tower of Hanoi problem.



*Example of what I mean in code*

**The puzzle should be able to solve itself**

My version of the Tower of Hanoi game has no user input in regards to moving the disks, the game finishes itself by use of the algorithm shown on the screenshots above. The function “towers” determines where a disk should move depending on its value and on which peg it currently is. The function “moveDisk” then moves the disk to its predetermined location.

**No exemplars**  
The biggest reason for this is that I started working on this assignment a bit too late, which resulted in me only being able to get the bare minimum done before the review.