Tower of Hanoi

Generated by Doxygen 1.8.13

Contents

Chapter 1

Tower of Hanoi Library

1.1 Game Definition

It consists of a finite set of **Rods** and **Disks**. The rods are organized in a line and each disk has a different diameter.

- **Start:** All the disks are organized in ascending diameter in the first rod.
- **Goal:** Move all the disks to the last rod.
- **Rules:**
  1. Only one disk can be moved at a time.
  2. Only the disks on top of each rod may be moved.
  3. Disks cannot be placed on top of smaller disks.

1.2 How to Use

The representation of a Tower of Hanoi game is made by the structure TowerOfHanoi. The first step is to create an instance of that structure and initialize it. After that, the movements can be executed and, at any point, the end of the game can be checked.

All the game manipulation functions are available in the file TowerOfHanoi.h

1.3 Example - Creating and Solving a 3-disk Game

```c
#include "TowerOfHanoi.h"

int main(void)
{
    unsigned numberOfDisks, numberOfRods;
    MoveError moveStatus; /* Unused here */
    TowerOfHanoi th;

    /* Initialization */
    numberOfDisks = 3;
    numberOfRods = 3;
    initializeTowerOfHanoi(&th, numberOfDisks, numberOfRods)

    /* Solve Game */
    moveStatus = moveDisk(&th, 0, 2);
    moveStatus = moveDisk(&th, 1, 1);
    moveStatus = moveDisk(&th, 0, 1);
    moveStatus = moveDisk(&th, 2, 2);
    moveStatus = moveDisk(&th, 0, 0);
    moveStatus = moveDisk(&th, 1, 2);
    moveStatus = moveDisk(&th, 0, 2);

    return gameOverTowerOfHanoi(&th);
    /* 0 (true) because the game is over */
}
```

1.4 Interactive Game

It is implemented an interactive game. After building, execute `build/bin/InteractiveTowerOfHanoi`.

```
  0 1 2
0 X - -
1 X - -
2 X - -
number of moves:   0

Select a disk to move:
```

Above is exemplified the execution of the app. You may insert which disk to move and to which rod it should be moved. The game will change its state depending on whether a valid move was inserted or not.

When the game is finished, it prompts a congratulation message and tells you in how many moves you completed the game.

```
Congratulations, you have finished the game in 7 moves.
```

1.5 Optimized Play

There is also an app which solves the problem using recursion. First define a sub-tower as a Tower of Hanoi made of a smaller number of disks. Then the optimized algorithm used is as follow:

- If the disk to be moved is the smallest, just move it.
- If the disk to be moved is not the smallest:
  1. Move the sub-tower over the selected disk to the other rod.
  2. Move the disk to the desired rod.
  3. move the previous sub-tower onto the selected disk.

With such algorithm, the problem is solved requiring to move the biggest disk to the last rod.

The app shows in the standard output all the steps to solve the problem with the minimum number of moves.

```
  0 1 2
0 X - -
1 X - -
2 X - -
number of moves:   0

  0 1 2
0 - - X
1 X - -
2 X - -
number of moves:   1


  0 1 2
0 - - X
1 - X -
2 X - -
number of moves:   2


  0 1 2
0 - X -
1 - X -
2 X - -
number of moves:   3
```

```
  0 1 2
0 - X -
1 - X -
2 - - X
number of moves:   4


  0 1 2
0 X - -
1 - X -
2 - - X
number of moves:   5


  0 1 2
0 X - -
1 - - X
2 - - X
number of moves:   6


  0 1 2
0 - - X
1 - - X
2 - - X
number of moves:   7


Congratulations, game finished in 7 moves.
```

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 TowerOfHanoi Struct Reference

Representation of the Tower of Hanoi game.

```
#include "TowerOfHanoiLib/TowerOfHanoi.h"
```

Data Fields

- unsigned **numberOfRods**
- unsigned **numberOfDisks**
- unsigned **numberOfMoves**
- bool **position** [MAX_NUMBER_OF_DISKS][MAX_NUMBER_OF_RODS]
- char **asString** [MAX_NUMBER_OF_CHARS]

4.1.1 Detailed Description

Representation of the Tower of Hanoi game.

Rods Representation: Each rod is represented by a column in the *position* matrix. The starting Rod is the column zero and the ending Rod is the last one. The number of Rods is set dinamically and can be accessed by the member *numberOfRods*.

Disks Representation: Each disk is represented by a line in the *position* matrix. Disks with small diameter are on top. Disks with larger diameters are at the bottom. When the disk *i* is on the Rod *j*, the value *i,j* of the *position* matrix is *true*. It is *false* if the disk is not there. The number of Disks is set dinamically and can be accessed by the member *numberOfDisks*.

Move Counter: After each move the member *numberOfMoves* is incremented being, therefore, the counter of moves.

String Representation: The game may be displayed as a string. In that case, the member *asString* has to be updated and can be shown using, for example, *printf* (from stdio.h).

Note

The implementation uses static allocated data. Its main purpouse is to be played or watched by humans so long games are not that useful. Dinamic memory allocation could be used though (it is not difficult to implement).

See also

`Tower of Hanoi wikipedia page`.
`Play the game online`.

The documentation for this struct was generated from the following file:

- TowerOfHanoiLib/include/TowerOfHanoi.h

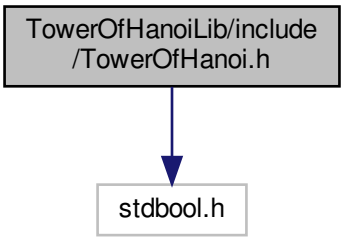
Chapter 5

File Documentation

5.1 TowerOfHanoiLib/include/TowerOfHanoi.h File Reference

```
#include <stdbool.h>
```

Include dependency graph for TowerOfHanoi.h:



Data Structures

- struct TowerOfHanoi

  Representation of the Tower of Hanoi game.

Macros

- #define **MAX_NUMBER_OF_RODS** 3u
- #define **MAX_NUMBER_OF_DISKS** 9u
- #define **MAX_NUMBER_OF_CHARS** (2∗MAX_NUMBER_OF_RODS + 2) ∗ (MAX_NUMBER_OF_DIS↩ KS+1) + 22

Enumerations

- enum MoveError {
  valid_move, **invalid_disk_error**, **invalid_rod_error**, **disk_not_on_top_error**,
  over_smaller_disk_error, **no_move_done_error** }

  Representation of all possible outputs of a move.

Functions

- void initializeTowerOfHanoi (TowerOfHanoi *const th, const unsigned numberOfDisks, const unsigned numberOfRods)

  Initialize a Tower of Hanoi with all the disks at the beginning.
- bool gameOverTowerOfHanoi (const TowerOfHanoi *const th)

  Check whether the game is over or not.
- MoveError moveDisk (TowerOfHanoi *const th, const unsigned diskToMove, const unsigned targetRod)

  Move a disk in the game if possible. It returns the movement status.
- void towerOfHanoi2string (TowerOfHanoi *const th)

  Update the string representation of the input game.

5.1.1 Enumeration Type Documentation

5.1.1.1 MoveError

```
enum MoveError
```

Representation of all possible outputs of a move.

There is a defined set of errors when attempting to move a disk, named:

- **valid move**: the requested movement does not break any rule.
- **invalid disk error**: the selected disk does not exist.
- **invalid rod error**: the rod to where the disk has to be moved does not exist.
- **disk not on top error**: the selected disk is not on the top of its rod.
- **over smaller disk error**: the disk is being moved over a smaller disk.
- **no move done error**: the selected disk is at the selected rod.

All of them, including a success move, are represented in that *enum*.

5.1.2 Function Documentation

5.1.2.1 gameOverTowerOfHanoi()

```
bool gameOverTowerOfHanoi (
            const TowerOfHanoi *const th )
```

Check whether the game is over or not.

Parameters

in	*th*	Tower of Hanoi game to be checked.

Returns

A bool value indicating if the game is over.

5.1.2.2 initializeTowerOfHanoi()

```
void initializeTowerOfHanoi (
            TowerOfHanoi *const th,
            const unsigned numberOfDisks,
            const unsigned numberOfRods )
```

Initialize a Tower of Hanoi with all the disks at the beginning.

Parameters

in,out	*th*	Tower of Hanoi structure.
in	*numberOfDisks*	
in	*numberOfRods*	

5.1.2.3 moveDisk()

```
MoveError moveDisk (
            TowerOfHanoi *const th,
            const unsigned diskToMove,
            const unsigned targetRod )
```

Move a disk in the game if possible. It returns the movement status.

Parameters

in,out	*th*	Tower of Hanoi structure.
in	*diskToMove*	The number of the disk to be move.
in	*targetRod*	The rod to where the disk will be moved.

Returns

MoveError Status of the move. See MoveError for more details.

5.1.2.4 towerOfHanoi2string()

```
void towerOfHanoi2string (
            TowerOfHanoi *const th )
```

Update the string representation of the input game.

Parameters

in,out	*th*	Tower of Hanoi structure.

Index