

GA Project Planning

Version : 8

Date : 19th Feb 2024

Choice of game: tower of Hanoi

Wireframe

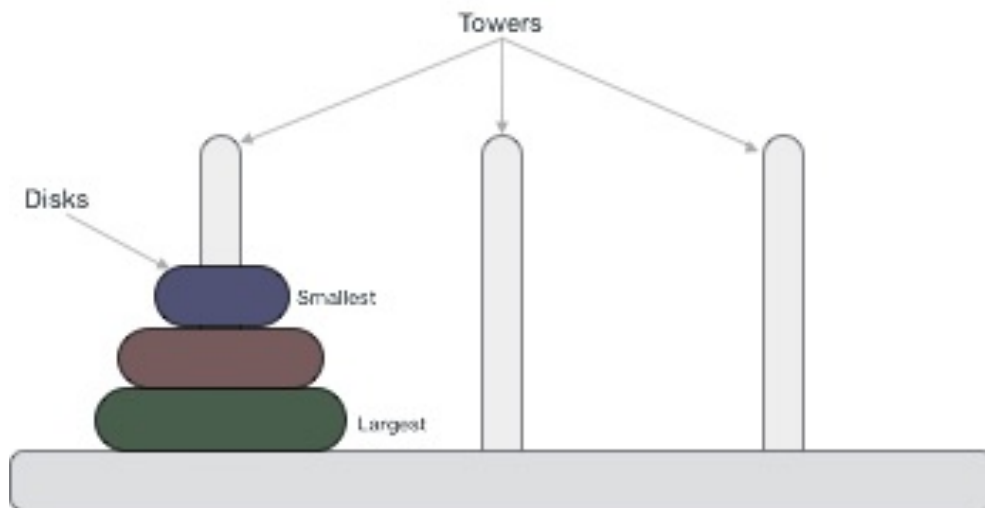


Figure 1: Tower of Hanoi [1]

- there tower will be created
- Multiple disc will be created
- A base plate will be created
- Rules:
 - Disc transferred from one tower to another tower, it cannot be bigger than the designated top disc of that designated tower.
 - One disc can only be moved at one time
 - Disc to be moved to a new tower from new tower with all disc from smallest on top to largest at the bottom of the new tower.
- Input
 - Numeric keypad to be used
 - Only 1,2,3 on keypad
 - Usage example: press 1 then press 3 to move the top disc of tower 1 to tower 3, doing all the above mentioned "Rules" check

Code

- a base class will be created for tower and disc
- 3 instances of tower and multiple instance of disc will be created
- Three queue array for tower to keep information of disc on each tower.
- When top disc is taken from certain tower, do a pop() function call from origin tower and a push function call to the designated tower, while doing all the business logic check (Rules check).

User

- will be a single user game

Timer:

- if time allows, a timer User interface will be shown, with the movement of the first disc to trigger start time. And correct assembly of new tower as end time.

Number of Move Display:

- if time allows, record number of moves as a variable and display on a UI on screen.

Note:

- automated running of game using recursive will NOT be added.

Reference:

[1] Picture taken from https://www.tutorialspoint.com/data_structures_algorithms/tower_of_hanoi.htm

Appendix:

Pseudocode:

- 1) Declare empty tower1, tower2 and tower3
- 2) Declare number of disc into variable
- 3) The Data update will have to update the html UI
- 4) Add an event listener for number keypad entry
 - 1) If key ==1
 - 1) Check if from tower variable IS NOT -1
 - 1) Set totower to 1
 - 2) Else
 - 1) Set fromtower to 1
 - 2) If key ==2
 - 1) Check if from tower variable IS NOT -1
 - 1) Set totower to 2
 - 2) Else
 - 1) Set fromtower to 2
 - 3) If key ==3
 - 1) Check if from tower variable IS NOT -1
 - 1) Set totower to 3
 - 2) Else
 - 1) Set fromtower to 3
 - 4) If totower !=-1
 - 1) Move tower
 - 2) Reset fromtower and totower variable
 - 5) Check winning on alternate tower?
- 5) Add IsTowerEmpty check
 - 1) Use length==0 of tower array to check for emptiness
- 6) Add [check tower is it winning] check
 - 1) Currently assuming tower 3 is the tower to be check (actually tower 2 also can check)
 - 2) First check for length of tower is it same as number of disc, if not, return false
 - 3) By right, number 1 above will suffice as the rules will be check before allowing for stacking to tower.
- 7) Declare disc color array
- 8) Populate disc and color
 - 1) For loop the number of disc
 - 1) Create element
 - 2) Classlist add 'disc' from style.css

- 3) Load background color to disc color array
- 4) Use index of for loop to control reducing size of disc
- 5) Push each disc into a dictionary array to hold all disc
- 9) Grab all tower html using `querySelectorAll(.tower)`
- 10) Add `insertDisc` to tower function
- 11) Add `removeDisc` from tower function»