2048 FIBONACCI

Dulipudi Naga Praharshini: 21b01a0545:CSE
Amjuri Veeramani Syamala Devi: 21b01a0505: CSE
Chalumuri Harshita: 21b01a0412: ECE
Kanna Naga Parvathi:21b01a0116: CE
Dalavi Vasudha: 21b01a0108: CE

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN

28th January,2023

Introduction

- The Fibonacci version of the 2048 game! Move tiles by swiping to any direction. Tiles with consecutive Fibonacci numbers will merge, get to the 2584 title to win! Each Fibonacci number is the sum of the previous two numbers: 1, 1, 2, 3, 5, 8, 13,.....2584

Approach

- Firstly we analyze the problem statement,and later on,we fetched the sources from google and other websites and we also add some logic that we learned from python programming in our academics and from google by combining both.we created a code for the problem statement.
- For the execution of the code we came up with different platforms like python online compiler and pycharm.
- As we done the execution in python compiler we also tried in pycharm but we faced some issues.
- Issues like update , grid, display function and we tried to built the code again with all the constrains.
- Later,after all changes we used python Idle to run the code.

Learning

Python has a built-in module that you can use to make random numbers.
The random module has a set of methods:

- randrange(): Returns a random number between the given range
- randint(): Returns a random number between the given range
- random(): Returns a random float number between 0 and 1
- setstate(): Restores the internal state of the random number generator
- getrandbits(): Returns a number representing the random bits

Challenges

- First we struggled for understanding about the problem statement.
- To Overcome this problem we searched various websites to get proper information about the problem.
- We took the code from various websites and practically tested it in IDLE'S like pycharm.
- Unable to execute the code in Pycharm.so we used python IDLE.

Statistics

Number of lines of Code : 179

Number of functions : 8

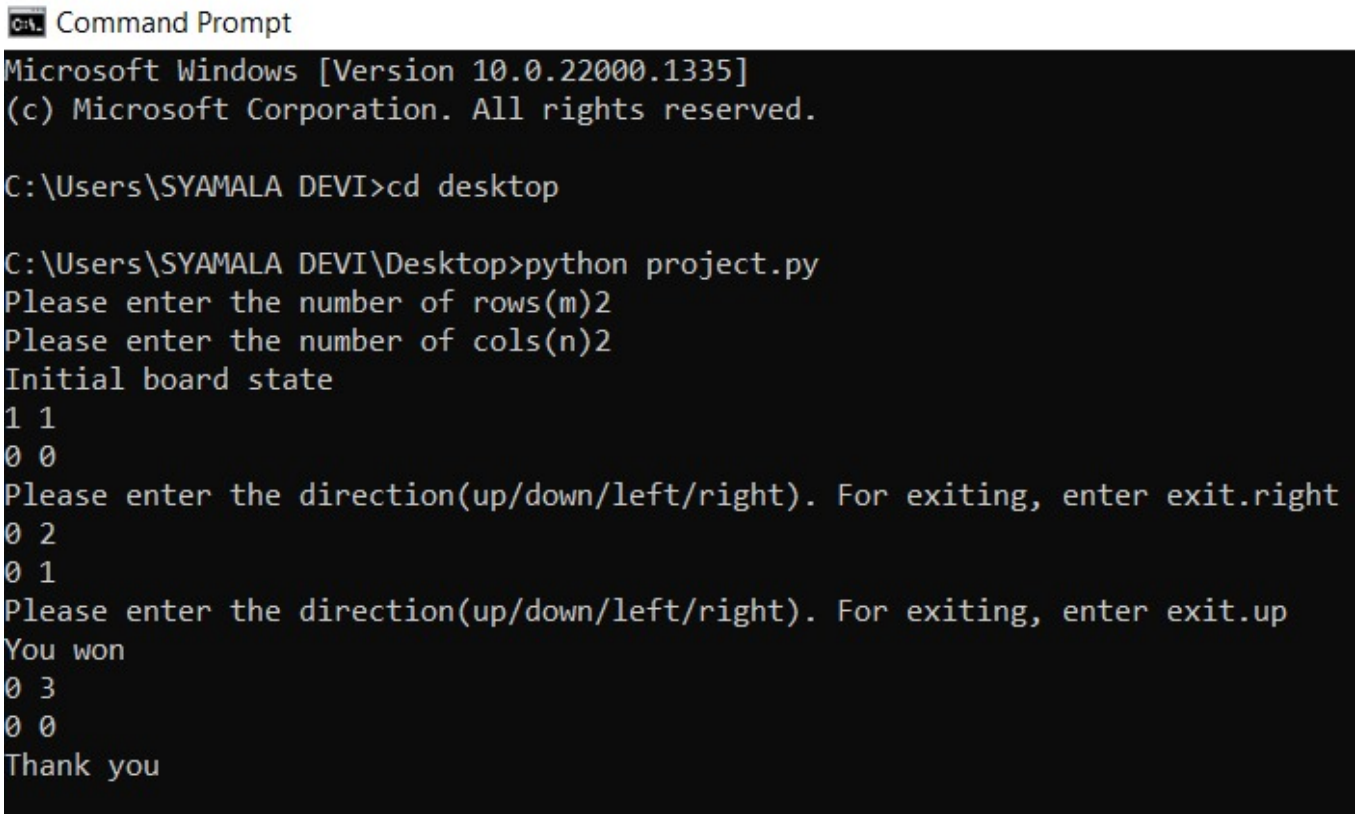
- The fill randpos() function takes in a game board and a number of random positions to fill with the value 1.
- The generate board() function creates a new game board with a given number of rows and columns, filled with the value 0.
- The generate fib cache() function generates a cache of the Fibonacci series up to a given number of terms and creates a map of those terms to their position in the series.
- The print board() function simply prints out the current state of the game board.

- The get sum list() function takes in a list of values and a Fibonacci cache, and returns a new list of the sums of adjacent elements in the input list that are also in the Fibonacci series.
- Finally, the update board() function takes in a game board, a specific row or column to update, a list of values to update it with, and a direction (up, down, left, or right) to update it in

Demo

- To run a demo of this code, you would need to first create a game board by calling the generate board function and passing in the desired number of rows and columns.
- You would then need to generate the fibonacci series and map by calling the generate fib function and passing in the same number of rows and columns.

Screenshots



Thank You