ReDI Python I: Maze Game

02.10.2017

Overview

This is a proposal for a course project in Fall 2017 Python Basics course at ReDI School

Goals

1. Learn how to build simple game with pseudo-graphics interface
2. Understand data structures like dictionaries and two-dimensional arrays
3. Communicate with fellow developers, define modules and distribute tasks in the group. Combine the pieces together

Specifications

The idea is to write a maze game, where the player defines the size of the maze to be generated by the game and then tries to escape it by moving right, left, up or down.

Basic version (single player)

The basic version will include square shaped labyrinth (a grid with some cells marked as walls) and one player. You start at entrance at the border and try to exit pressing arrow buttons on the keyboard to make moves. After you escape the game should display number of steps and time used.

Extended version (two players)

Extended version might allow multi-player setup with two players sharing the same keyboard (one uses arrow keys and another one using keys ‘a’, ‘d’, ‘w’, ‘s’ to move left, right, up and down respectively. When one player escapes, the game offers to repeat. Total score (number of wins) is tracked.

Milestones

One Street Game

Write a program that takes from the user 'Right' or 'Left'. The program should print accordingly

```
    an X showing where you are on a 10 block street.

    --------------------------------

    Initially Program Prints:

    X . . . . . . . . .

    If the user enters 'Right', program prints:

    . X . . . . . . . .

    If the user enters another 'Right', program prints:

    . . X . . . . . . .

    If the user enters 'Left', program prints:

    . X . . . . . . . .
```

A Walk in the park

Write a program that takes from the user 'Left', 'Right', 'Up', 'Down'. The program should print accordingly and X showing where you are on a 3X3 Grid. Initially player is in top left corner

```
  ---------------------------

  Initially Program Prints:

    X . .

    . . .

    . . .

  If the user enters 'Right', program prints:

    . X .

    . . .

    . . .

   If the user enters  'Down', program prints:

    . . .

    . X .

    . . .
```

```
If the user enters 'Left', program prints:

. . .

X . .

. . .

If the user enters 'Left', program prints:

. . .

. . .

. . .

'You are no longer in the park! Good bye!'
```

Wall

Write a program that takes from the user 'Left', 'Right', 'Up', 'Down'. The program should print accordingly

and X showing where you are on a 4X4 Grid. The 4X4 Grid has blocking walls , that you can't walk through it

The position of these walls you take as input.

Initially player is in top left corner

```
---------------------------

Initially Program Prints:

X . . .

. W . .

. . W .

. . . .


If the user enters 'Right', program prints:

. X . .

. W . .

. . W .

. . . .


If the user enters  'Down', program prints:

. X . .

. W . .
. . W .

. . . .

'Wall!'
```

If the user enters 'Right', program prints:

```
. . X .

. W . .

. . W .

. . . .
```

If the user enters 'Down', program prints:

```
. . . .

. W X .

. . W .

. . . .
```

Maze 0.1

Write a program that takes size of maze if it is 5, the maze is 5 X 5 , and a list of wall positions, and play maze!

Create a class called Board:

- Board Object is initialized given two parameters: size, is_random
- if is_random is true, block positions are generated
- if is_random is false, program asks user to enter wall positions

You can do the following actions on board:

1. add wall
2. add player
3. remove player
4. check empty cell