The Maze Adventure

An old, legendary Treasure Hunter has contacted you for help to more efficiently train for big, grand adventures. Dungeon crawling just isn't what it used to be, and it's becoming increasingly difficult to stay relevant and competitive in the modern treasure hunting space. A technological breakthrough is needed!

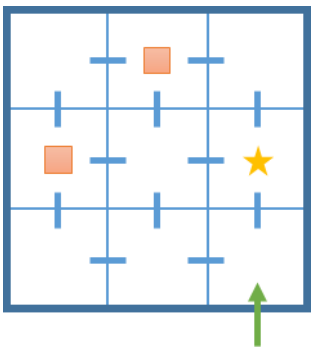
Your mission is to assist in extending an existing virtual Maze adventure training application by adding on **capability to generate mazes**.

Mission

Create an API that can generate random mazes for the Treasure Hunter to practice on. Your solution will be plugged into an existing maze simulator, in which a player navigates a maze in search of the big treasure. The Treasure Hunter is looking for a robust, extensible and clean solution that follows a minimal list of functional requirements (below) – but outside of that, your creativity is much appreciated!

The maze you generate should have a layout similar to the one in the picture, where:

- Green arrow = entrance
- Red square = trap
- Star = big treasure
- Short line = door/passage
- Thick edge = maze wall



Example of a 3x3 maze

Constraints

The Treasure Hunter has been dreaming up some requirements for this first iteration:

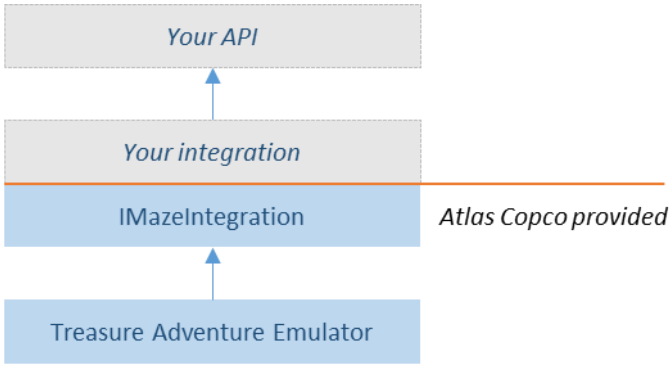
- Mazes can be of varying size, but they are always square (e.g. 3x3).
- It should be possible to generate different *types* of mazes with varying behavior at some point. But for now, a "**VerySimpleMaze**" is enough.
- A Maze consists of **rooms** and an **entrance** (a starting room).
- Each **room** is connected to adjacent rooms, but the edge of the maze cannot be traversed.
- Each **room** has a description, which is composed from its **type** and potential **other actions**.
- When a new maze is generated, all maze entities (rooms, entrance, room properties) are randomized so that each new maze is different.

- There are multiple types of rooms (feel free to extend), each with their own imaginative descriptive texts:
    - Forest
    - Marsh
    - Desert
    - Hills
    - Example of description: *You are looking at a heap of leaves and feel slightly elated.*
- Each type of room may have **varying behavior**. Only the following few **traps** are needed for now, but more are to be expected:
    - **Marshes** have a 30% chance to cause the player to sink when visited (ending the adventure).
    - **Deserts** have a 20% chance to cause dehydration when visited (ending the adventure).
    - If a behavior is triggered, the final room description will be affected according to: `room description + behavior modifier`.
- One single room in the maze has a **treasure**. When found, the adventure will end.
- The treasure room can never have a trap.
- The entrance room can never have a trap, and cannot have a treasure.

Integration

In order for the Treasure Hunter to integrate your maze generation API in a standardized way, you are also asked to implement a simple façade, **IMazeIntegration**. This interface is provided for you.

The Treasure Hunter will take your version of this façade and plug it into its patented *Treasure Adventure Emulator*, from where virtual adventures and great joy can be had. This will be used for verification purposes.



Solution overview

Technicalities

Develop your solution using:

- The C# language
- Any version of Visual Studio 2015 or later
- .NET Framework 4.6.2
- References only via NuGet

Deliverables:

- All source code in a single solution file.
- Separate DLL for `IMazeIntegration` + its implementation, i.e. *only* exposing types necessary for integrating with your API.