

SCP Maze

Program Documentation

by elidacon

0.1 Introduction

This program is developed using C++17 and follows the principles of object-oriented programming. It employs Model-View-Controller (MVC) design pattern. The graphical user interface (GUI) implementation is based on QT 6.4.3.

The 'Maze' program features two interactive tabs: 'Maze' and 'Cave'. Each tab serves different purposes and offers various functionalities.

On the 'Maze' tab, users can interact with the application to manage mazes. There is an 'Open' button that allows loading a pre-existing maze from a file. Additionally, there are fields where users can specify the dimensions of the desired maze. To create a new maze, users can click the 'Generate' button, which will construct a maze according to the specified size. Once satisfied with the generated maze, it can be saved using the 'Save' button. Users also have the ability to 'solve' the maze, meaning to find a path from one point to another. They can designate a starting point by clicking the left mouse button and mark the destination with the right mouse button directly on the maze layout. Once the endpoint is established, the software will visually trace the route from the starting point to the designated end point.

Switching to the 'Cave' tab, a similar interface is presented, albeit with some unique features specific to caves. Users can again load a pre-existing cave configuration from a file via the 'Open' button. Fields allow the input of the dimensions of the cave and a chance value for initializing cells. The 'Generate' button creates a cave based on these parameters, and the 'Save' button stores the current state of the cave.

Additionally, the 'Cave' tab includes two sliders for birth and death limits, each offering a range from 0 to 7. These sliders control the parameters of the cellular automaton algorithm applied within the cave.

The 'Cave' tab also features two action buttons: 'Next step' and 'Play.' Pressing 'Next step' advances the cellular automaton algorithm to its next stage, allowing users to observe the evolution of the cave one step at a time. The 'Play' button initiates a continuous rendering of all iterations of the algorithm, with the speed of progression determined by a delay. A delay can be set using a dedicated field, which determines the pause between each iteration of the algorithm.

0.2 Usage

To install the program, you first need to navigate to the directory containing the program's source files. Run 'make install'. After the installation process is completed, you can start the program by running either 'make open' or './install/scp_Maze'

0.3 Available buttons

Maze tab:

Open	Open a .txt file, containing a maze
Size (RxC)	Set the size of a maze to be generated
Generate	Generate a maze
Save	Save current maze

Cave tab:

Open	Open a .txt file, containing a cave
Size (RxC)	Set the size of a cave to be generated
Size (RxC)	Set the chance for the starting initialization of a cave's cell
Generate	Generate a cave
Save	Save current cave
Birth limit	Birth limit for cellular automaton algorithm, range 0-7
Death limit	Death limit for cellular automaton algorithm, range 0-7
Delay	Delay in ms between rendering iterations for the Play button
Next step	Render the next iteration of the cellular automaton algorithm
Play	Starts to render iterations of the cellular automaton algorithm