Maze game

09 April 2020

19:02

Technical skill

* Stack
* Queue
* Binary file read and write operations
* Simple Object oriented programming
* Simple mathematics
* Imported external dll file

Stacks

* Used in the maze maker to make the forward and backward buttons in your case you used save stack as the backward button.
* The save stack allows you to push all of the walls to the stack and when time to save you just keep popping from the stack till it's empty.

Queue

* The queue is used for the level loading.
* You checked the directory for all of the files and then add them to a queue.
* When it come time to load each level they will be loaded a curtain order as the names are in order and the queue allows you to keep that order.

Binary Files

* The binary you are using is a csv file, they keep data by have a comma after everything so we can use that to are advantages as is can be read as coordinates or dimensions for a shape.
* You have to read and write to the file in a curtain way though other wise the data won't be interpreted the right way.

Simple oop

* The main example of simple object oriented programming is the wall, Character and level classes.
* The MazeViewModel and MazeMakerViewModel could be argued to be complex object oriented programming instead of simple as there is lot more to them like inheritance as your are using a package call Caliburn.micro and all view models have to inherit the Screen or Conductor classes.

Simple Maths

* In the MazeMakerViewModel when adding a wall a simple calculation is made to find the right coordinate for the rectangle to be drawn from as the rectangle is drawn from the top left corner.
* In the SetCusor method there is simple to math to set the user cursor relative to the windows position.

Imported Dll

* The imported dll is the user32 dll native to windows inside this dll is the SetCursorPos method that we can using in C# to allow us to set the users cursor position.

Caliburn.micro

* Caliburn.micro is a package used in WPF to help users set up a MVVM model (Model View, view, Model).
* The core use of the package is so that the view (window) will be updated every time a variable is bound to the view eg the player position.