



# Introduction to Programming

## D Task 4.4: Maze Creation

### Overview

This program will allow you to use the Gosu cycle to create a simple maze on the screen.

**Purpose:** Modify a Ruby program which uses references.

**Task:** Modify the provided code in the Gosu window *initialize()* procedure(method).

**Submit To:** Canvas when complete

**Time:** This task should be started in your fourth lab class and submitted for feedback before the start of week 11.

**Resources:**

[Sobkowicz, M 2015 \*Learn game programming with Ruby : bring your ideas to life with Gosu\*. The Pragmatic Bookshelf \(See chapter 7 for help the grid aspect of the Maze Task\)](#)

[Gosu Ruby Documentation](#)

[Gosu site](#)

[Gosu game video tutorial](#)

### Submission Details

You must submit the following files to Canvas:

- Basic *gosu\_maze\_creation.rb* source code
- Screenshot of the window showing the execution of your program.

Make sure that your task has the following in your submission:

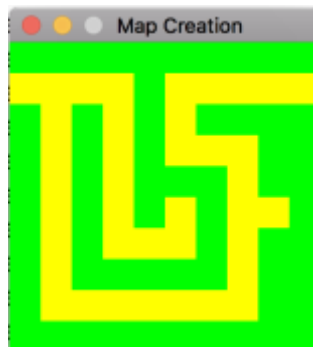
- The program must allow the user to create a maze on the screen by modifying code in the appropriate sections.
- Code must follow the Ruby coding convention used in the unit (layout, and use of case).
- The code must run and the screenshot show it working.
- This program does NOT need to have a procedure for main – the Gosu cycle is the main cycle.
- Your program must have the indicated local variables and use them appropriately.

## Instructions

Use the code provided (from this task's resources in Canvas) to get started.

You must complete the code so that it works as follows:

1. The code in the *initialize()* procedure of the Gosu window should be completed to set up cells that are connected to each other with variables joining each cell to its neighbours (using references).
2. The user should be able to left click on cells on the screen to create mazes (and later in the Maze Search task we will use recursion to find a path through the maze).
3. Each cell clicked on should turn yellow.



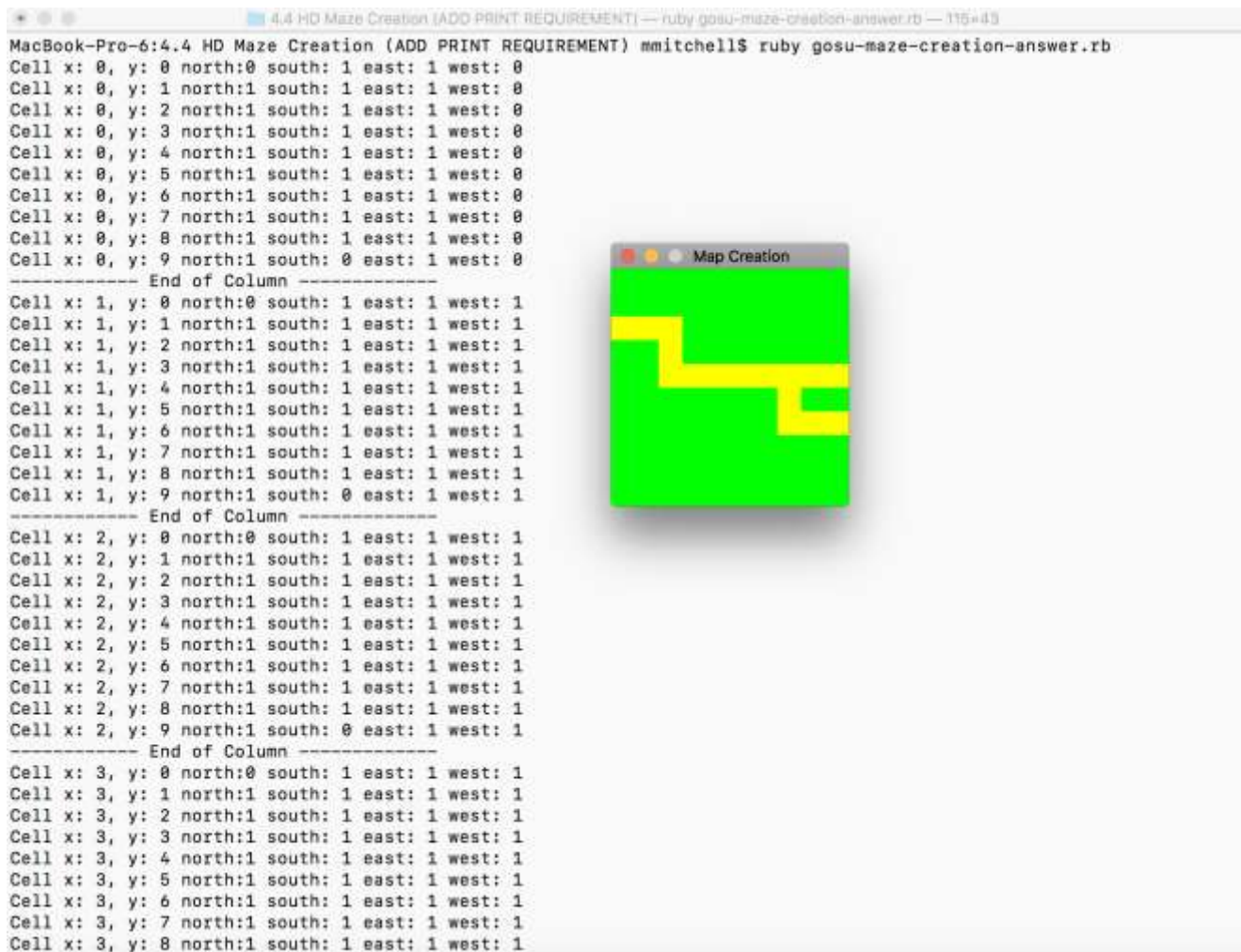
4. Once this is working (or perhaps before) add code to print out each cell and indicate whether the reference to the neighbor on each side is nil or not, as per the following:

```
Cell x: 0, y: 0 north:nil south: 1 east: 1 west: Nil
```

In this case there is no neighbor to the north or the west.

(NB: whether the east or the west neighbor is nil will depend on your perspective – i.e. is your perspective looking into the screen, or out of the screen)

Your submitted screenshot should look something like the following:



```

MacBook-Pro-6:4.4 HD Maze Creation (ADD PRINT REQUIREMENT) mmitchell$ ruby gosu-maze-creation-answer.rb
Cell x: 0, y: 0 north:0 south: 1 east: 1 west: 0
Cell x: 0, y: 1 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 2 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 3 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 4 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 5 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 6 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 7 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 8 north:1 south: 1 east: 1 west: 0
Cell x: 0, y: 9 north:1 south: 0 east: 1 west: 0
----- End of Column -----
Cell x: 1, y: 0 north:0 south: 1 east: 1 west: 1
Cell x: 1, y: 1 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 2 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 3 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 4 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 5 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 6 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 7 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 8 north:1 south: 1 east: 1 west: 1
Cell x: 1, y: 9 north:1 south: 0 east: 1 west: 1
----- End of Column -----
Cell x: 2, y: 0 north:0 south: 1 east: 1 west: 1
Cell x: 2, y: 1 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 2 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 3 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 4 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 5 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 6 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 7 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 8 north:1 south: 1 east: 1 west: 1
Cell x: 2, y: 9 north:1 south: 0 east: 1 west: 1
----- End of Column -----
Cell x: 3, y: 0 north:0 south: 1 east: 1 west: 1
Cell x: 3, y: 1 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 2 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 3 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 4 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 5 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 6 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 7 north:1 south: 1 east: 1 west: 1
Cell x: 3, y: 8 north:1 south: 1 east: 1 west: 1

```

***Also you should include code to print out your grid once it is created and  
Submit your completed code and screen shot to Canvas.***

End of Task