

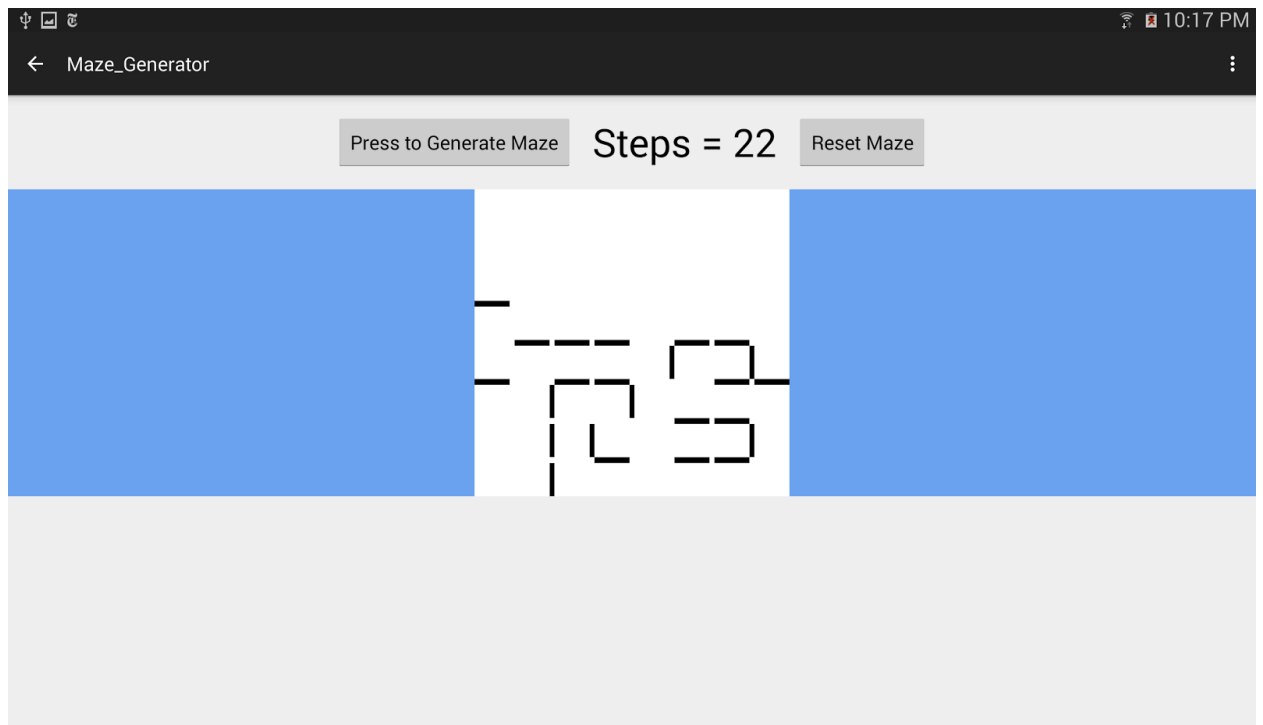
Yoav Nathaniel  
Artificial Intelligence  
Professor Kumar  
February 12, 2015

## Project 1 - Maze Generator - Technical Manual

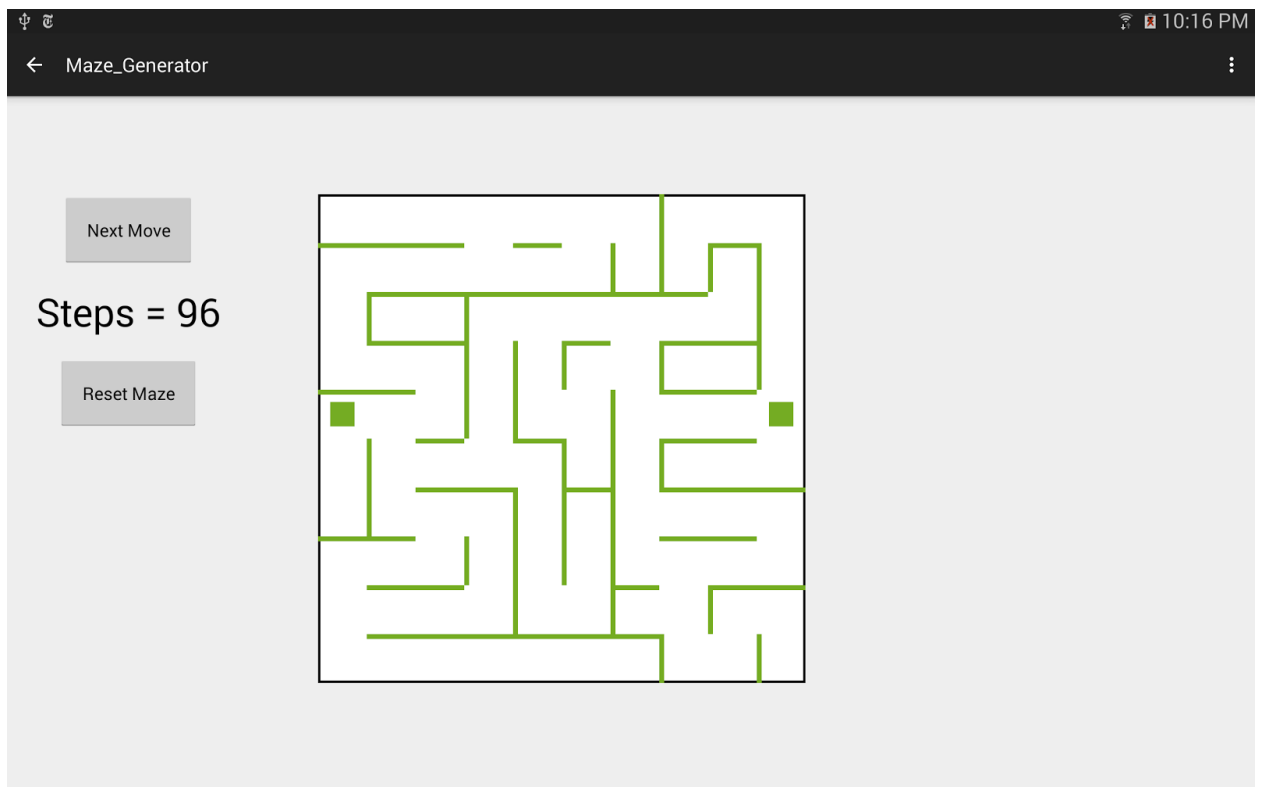
### **1 - Activities:**

IntroActivity - consider IntroActivity as the welcome activity. The program launches into it and it stands as the middle-man between all the different types of mazes. It allows the user to select the dimensions of the maze (using the spinner) and choose what type of maze the user wants to see. Descriptions for the choices are provided in the activity.

CompleteActivity - I first completed the project with a combination of textviews and LinearLayouts (really inefficient). Because this was completed first, I called it CompleteActivity. If the user chose an option for a maze with LinearLayout, they will reach this activity. It shows the maze, the amount of steps it took to generate that maze, and it has buttons to both allow you to reset the maze or generate another step/maze.



MainActivity - after extensive research and trial and error phase (of which I overcame), I managed to mimic the CompleteActivity, but using Imageview, Canvas, and Bitmap.



## **2 - AI Algorithms:**

The project generally builds the maze using Depth First Search. The Grid class contains all the Cells (which rooms or walls), and makes sure no walls are crossed, overwritten, other minor details. It travels through the 2D array of Cells using a stack and it keeps track of all of the visited cells.

The Depth First Search (explore function of Grid class) is activated every time another cells needs to be checked. This happens in both CompleteActivity and MainActivity.

Note: in order to maximize randomness of maze generation, the cells are pushed into the stack are at a random order as well.

## **3 - Bug Report:**

I do not suspect of any bugs in my code. I do, however notice that the maze sometimes closes of large parts of the grid, due to randomly built walls that close off the perimeters.

A light bug, might not be considered one, is that certain walls in the bitmap view don't fully touch. This is referring to where horizontal and vertical walls meet - sometimes their edges won't fully touch.

#### **4 - Feature Report:**

Regarding required features, I don't think the program lacks any.

The Maze Generator project features two different methods to draw the maze. It is left to user to decide which maze he'd rather see.

The LinearLayout changes in size according to the dimensions. The Canvas layout changed the size of the cells dynamically.

If I had more time, I would add another activity that uses the generated maze and gives the user a pacman figure to travel around.

#### **5 - Log:**

**1/28/15**

--Start maze generator project.

--Begin working on Grid and Cell classes using limits. Limits was an idea to keep track of an array of 4 booleans across all of the critical functions named below.

CheckLimits() would check around the cell and decide if it allows a wall to be possibly built around it. BuildWalls() and easyWalls() would use the limits set by CheckLimits() and randomly build walls around the cell. It would return the results to explore() and, with set order, check to see if the neighbors should be pushed to the stack.

--Critical functions built: checkLimits() (no longer exists), buildWalls(), easyWalls(), and explore() (elementary version).

**(Total: 2 hours)**

**1/29/15**

--Keep Cell class, but create Wall and Room classes (they extend the Cell class).

--Implement the BuiltFrom variable across the Grid, Room, and Cell classes. For any Cell that's a Room, I wanted to see where it came from, so the path is not ever blocked when I build walls.

**(1.25 hours)**

--Tested my ability to work with the different layouts and decided LinearLayout provided the greatest flexibility.

**(0.5 hours)**

--Created MapDrawer, MapRow, and MapBox classes. MapDrawer has one vertical LinearLayout filled with MapRow objects. MapRow has one horizontal LinearLayout filled with MapBox objects. MapBox has a textview which can be considered as a Wall/Room.

--Start working on MainActivity (things moved around and it is now considered CompleteActivity). Everything that's currently in DrawItAll() was in MainActivity.

**(2 hours)**

**(Total: 3.75 hours)**

**1/31/15**

--Designed the xml for activity\_main.xml. Only has the step count and the button to generate the maze.

--Improved MainActivity. The onCreate method has the button listener for the generate maze button.

--Built gridExplore() method in MainActivity. Correlated it to the button press. A big step because it is the first time I correlated MapDrawer/MapRow/MapBox classes to Grid/Cell class.

**(2 hours)**

--Built nextStep() method in MainActivity. Generated a guaranteed working and complete maze every time.

--Build resetGrid() method in Grid. Resets the Grid to make it like new.

**(0.5 hours)**

**(Total: 2.5 hours)**

**2/3/15**

--Build difficultWalls() in Grid class. An attempt to create a more difficult version than easyWalls(). It tried to solve a problem I call the Square. Where a maze is fully generated, yet it has an area with 4 or more cells completely open to each other. I had to build at least one wall in those 4 cells to create an actual maze out of them.

**(0.5 hours)**

--Experiment with canvas and bitmap in MainActivity. Try to build a much more efficient way to view generated maze.

**(0.5 hours)**

--Create IntroActivity and intro\_activity.xml. Designed an interface to launch to. The interface has buttons for two modes, Random Explore (to see the magic happen one step at a time) and Complete Maze (to generate a complete and working maze in one step). IntroActivity also has Radio Buttons to select the size of the grid.

--Add IntroActivity as the Launcher in the AndroidManifest. Make CompleteActivity a child of IntroActivity.

**(1.5 hours)**

**(Total: 2.5 hours)**

**2/4/15**

--Build hardWalls() in Grid class. A method that would work with easyWalls(). Before randomly building walls, I wanted to make as many turns as I could. I attempted to get rid of long straight lines with no turns. It was not successful.

**(0.5 hours)**

--Move MainActivity (referred to as CompleteActivity up until now) to CompleteActivity. Adjusted IntroActivity to work with CompleteActivity and MainActivity.

--Redo MainActivity, except use a Canvas and a Bitmap instead of the LinearLayouts. Created bitmapSurround() method to have black lines on the border of the canvas.

Created drawInterface() method to keep onCreate method more organized.

--Create ImageScreen class, extending ImageView. I wanted this class to fix the problem I had with the ImageView not updating. I used invalidate() but still didn't work. I ended up deleting and drawing the ImageScreen every time I wanted it to update.

--Create DrawCell class. An object used for drawing on the canvas.

**(2 hours)**

**(Total: 2.5 hours)**

**2/5/15**

--Attempt to reorganize my Grid class. Instead of building walls, I changed Grid's constructor to build all the walls in the beginning, and selectively take the walls down cell by cell. I wrote explore2() and newLimits() in Grid. Results were not successful, a tad confusing.

--Build paintTheRest() in CompleteActivity. Supposed to make the closed off and unreachable cells distinct and stand out.

--Fix issues with canvas.

**(Total: 2 hours)**

**2/7/15**

--Get rid of checkLimits() method and all of the wall building methods in Grid. Also throw away newLimits() and explore2().

--Write realWalls() method. I finally had in mind what I needed Grid to do. realWalls() combines an improved version of checkLimits() and all of my previous wall building methods. The biggest improvement is that instead of using booleans, I used ints to set a priority for building certain walls.

--To make things neater, I spread realWalls() into squareCheck() and priorityRandomBuild().

--Change the order of pushing cells into the stack in explore(). I decided to make it random.

**(Total: 1.5 hours)**

**2/11/15**

--Make a spinner in IntroActivity. Change the range of sizes available for the maze. Now it's from 6x6 to 14x14.

--Adjust the sizes of the cells in the canvas to become dynamic. The size of the cells depends upon the dimensions selected.

**(0.75 hour)**

--In the Random Explore versions of the maze, if the maze has finished generating and the goal cell hasn't been reached, a pop up alert comes up.

**(0.5 hour)**

--Add a reset button to both MainActivity and CompleteActivity. At any time, the user can reset the grid and the drawing of the maze.

--Take out paintTheRest(), doesn't look so good.

**(0.25 hour)**

**(Total: 1.5 hours)**

**(Total time for project: 18.25 hours)**