

CPE 102 Program 3 – Maze Game
Part 2 (Extra Credit)
Due: Monday 11/14/16

Suggestion: Finish your Lab7 before stating this part. The Lab will be a big help for this part.

Part 2: The Maze rendering will be as 10 points as extra credit to your projects. You need to demo on Monday 11/14 in the Lab. This is your only chance to demo and you will receive a grade based on what you have accomplished.

In Part 2 you will write code to have each class draw itself on the screen. *Do not work on the draw() methods of the Drawable versions of each of your classes until you believe your have their non-drawable versions working perfectly!!!*

A goal of your learning for this assignment will be how to *reuse class specifications through inheritance* and how to use a standardized *interface* for the drawing operation. To complete the program and its classes you will be required to use abstract classes, as well.

Part 2 files: Drawable.pde, Square.pde, Occupant.pde, Explorer.pde, RandomOccupant.pde, Monster.pde, Treasure.pde, Maze.pde, DrawableSquare.pde, DrawableExplorer.pde, DrawableMonster.pde, DrawableTreasure.pde, DrawableMaze.pde, and MazeGame.pde (whether or not you altered it)

Submit these files to the PolyLearn.

Part 2 Info

To use objects of the **Drawable** classes you design you will use a GameGUI class that manages the GUI (graphical user interface) and calls the methods of the **DrawableMaze** class to run the game. The GameGUI class is given to you in a Processing file named [MazeGame.pde](#). The GameGUI class has a draw method (called from the main Processing draw method) that will ask the **DrawableMaze** to draw itself and its occupants. It will also print a final message to the screen when the game is over. Calling methods of the Processing development environment is how the **DrawableMaze** and its **Occupants** get displayed on the screen. The **DrawableMaze** class will simply store an array of **Occupants** and call the draw method of each of them. The correct draw method for each **Occupant** object will be called because of *polymorphism*. That is, if the **Occupant** is actually a **Treasure**, then the **Treasure** class draw method will be invoked.

Drawing in the **Processing Development Environment** is done in terms of pixels in an x/y coordinate system (More explanation in the Lab7). The (0,0) pixel location is at the upper left corner of the screen. Positive x is to the right. Positive y is down.

The **Maze** is kept as an array of **Square** objects. Row 0 is the top most row. Column 0 is the left most column.



Note that pixels are in (x,y) and Squares are in (row,col). The pixel at (0,3) is on the far left of the screen, 3 pixels down from the origin. However, the Square at (0,3) is in the first row, but in the third column from the left of the screen. Keep this in mind when you do row/col to pixel conversion.

Okay, now what? How do I actually draw in the Processing environment? Once you have finished the “backend” of the maze game, you should begin to work on the `draw()` method of the Drawable versions. Lab7 will be an excellent introduction into the Processing Development Environment. Complete that lab before you attempt completing the drawable portion of this Project.

Solution Requirements

Converting to Processing

1. Download the given [MazeGame.pde](#) file and place it in a directory named MazeGame. You may alter this file in the one spot where it says CHANGE to give the explorer a name appropriate to your theme.
2. Download the given [MazeGenerator.pde](#) file and place it in the same directory. This file contains all the necessary code to generate a random maze each time you run your program.
3. Move all your code into the same directory (except for the P3TestDriver if you have it).
4. Change all your file extensions to .pde.

5. Finally, add font ([AmericanTypewriter-CondensedBold-48.vlw](#)) to your sketch by downloading the file. Then in the Processing window, click Sketch->Add File. Add the font to your sketch. It is used in the final scene.

Now you are ready to start drawing! You should be able to run the Processing sketch, which won't do a whole lot yet. If your Monster draw method is complete from Lab 7 you should see the Monsters moving around the screen.

Begin implementing the draw methods for each class that implements the Drawable interface and your Maze Game will start to come to life.