Graphical user interfaces, or GUIs, are a critical part of any modern application, including apps that run on phones, tablets, and televisions. Object-oriented programming plays a major role in developing graphical user interfaces, regardless of what programming language or technology is used. Using the object-oriented principles that we've learned, and by taking advantage of some already-existing tools, we can easily create programs with graphical user interfaces using Python.

Assignment

Your assignment is to create the 15 Puzzle (play it [here](link)) using Python and GTK. The assignment will consist of two sequential parts. For this first part, you are required to implement the graphical user interface for the game, which should include the following components and functionality:

1. A single window that is displayed when the program starts, which contains the entire user interface for the game.
2. A label to display the title of your game (different than the window’s title). You may be creative in naming your unique version of the game. The label’s text should be styled in a simple but unique way.
3. 16 tiles that appear as buttons in a 4 by 4 grid. The tiles (buttons) should be displayed in order, labeled 1 through 16, and styled in a simple but unique way. When one of the buttons is clicked, that button should become hidden, and any previously-clicked button should become visible again. Also, a message should be printed to the console that indicates that a tile was clicked, as well as the position of the tile that was clicked (see the sample section below for an example). This temporary functionality will prepare for part 2 of the assignment.
4. A “New Game” button that, when clicked, simply prints a message to the console indicating that this button was clicked.
5. A “Quit” button that, when clicked, causes the window to close and the program to terminate.
6. The window’s X button (in the corner) should likewise cause the window to close and the program to terminate.

For part 1 of the assignment, no additional functionality is required (e.g. shifting the tiles in order to play the game). You will implement the remaining features to make the game playable in part 2. You are welcome to continue working on additional features once you complete the requirements for part 1, but it is your responsibility to complete the requirements for part 1 of the assignment first, and submit it by the due date.

For this assignment, you should demonstrate the object-oriented principles that you have learned as you design and implement the classes that you will use to implement the graphical user interface and logic for the game. At a minimum, you should have separate classes for the user interface and game logic elements, and use aggregation to connect them. Keep in mind that the game logic for part 1 of the assignment will be minimal, but any necessary classes can be created now if desired.

Extra Challenges

- Create a game of a different size, such as 5 by 5, or allow the player to choose the size he or she wishes to play.
- Rather than using numbers to identify the individual tiles, display a rectangular division of a larger image inside each button. The objective of the game would then be to arrange the tiles in order to reveal the original picture.

Hints

- Refer to the class examples to see examples of how to use the various GTK widget classes and methods.
- Refer to the Python GTK+ API documentation and tutorial to understand the various classes and methods that are available to use.
- When creating colors, use a helpful tool to determine the RGB values. Here are two good options: color.adobe.com and colorpicker.com
Sample

An example running program:

Example console output for part 1:

A tile was clicked at position (0, 0)
A tile was clicked at position (2, 0)
A tile was clicked at position (1, 1)
A tile was clicked at position (3, 1)
A tile was clicked at position (0, 2)
A tile was clicked at position (2, 2)
A tile was clicked at position (1, 3)
A tile was clicked at position (3, 3)
New Game clicked