

General

At the top of the every document that you create (word processing files or source files) include:

```
/**
 * Description of the class or document.
 *
 * @author YOUR NAME
 * @version CS162 Lab #, mm/dd/yyyy (replace with the last edit date)
 */
```

Submit in this lab via Moodle using the “Lab #” link. Your assignment must be uploaded by the assigned due date and time. Moodle will automatically close the link at that time. It is strongly highly recommend you do not wait until the very last minute to submit your work.

Concepts

This lab will give you an introduction to the major concepts in Java GUI programming with Swing. You will get experience with Java panels, GUI controls (buttons, menus, etc.), text display, image display, and most importantly the Swing event handling structure. This project will also further your experience in code refactoring.

Background

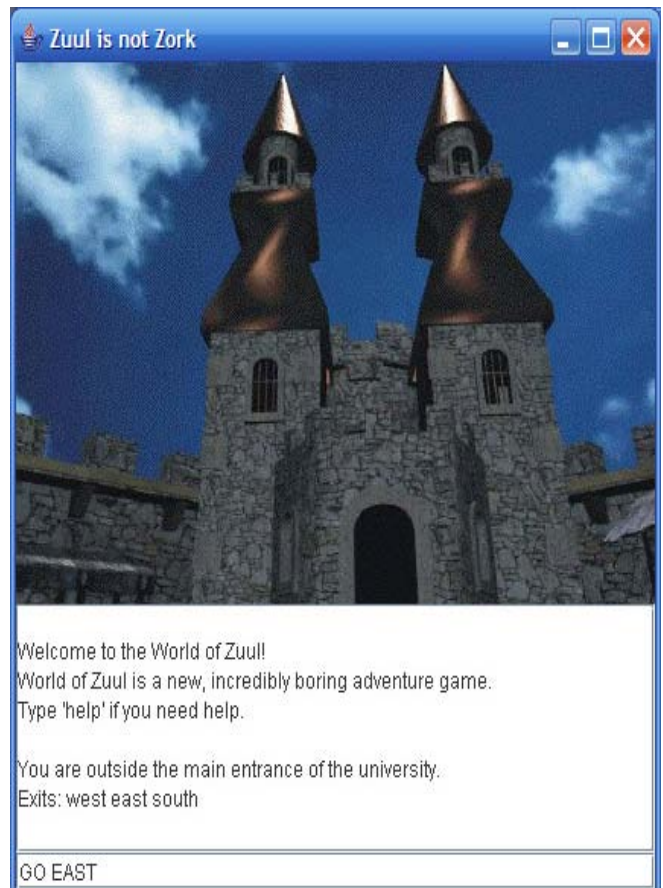
Read chapter #11 and review the online tutorials on GUI Programming. Complete and review the pre-lab assignments.

Assignment

Zuul with GUI

Your task is to add a graphical user interface (GUI) to your version of the Zuul game. Include the “transporter room” logic that you completed earlier this term. Feel free to re-factor the architecture of the code to allow easier implementation of the GUI and any features that you choose to add. My strong suggestion is to bite off little bites at a time, get something working and tested, then go on to the next bite! Here are some suggested chunks:

1. Re-factor all input and output to a new GUI class (no GUI code for this step, just isolate all I/O into this new class).
2. Create the frame, GUI controls (Image label, text boxes, menu). Make visible, but no event handlers yet.
3. Get event handlers for Menu operational (exit, help, about).
4. Get the text to display and scroll in the text display box.
5. Refactor the “play” method in Game to remove the loop; the play loop is replaced with the event loop in the Java AWT/Swing classes.
6. Add the event handler for the text input



to read the command string and send it to the input object (this is hard as you have to reverse some of the logic in the Game class).

7. Add a new field to the Room class that contains the relative location of the picture image for each room. Display the picture on each room/location change using a JLabel image icon.
8. Add any of the challenge features that you want to tackle.

Requirements

Your implementation must fulfill the following minimum requirements:

- The application runs outside BlueJ (thus it has an entry point).
- It opens its own frame.
- Each location must have an associated image.
- Locations still have text descriptions.
- The application has a menu. Menu items are at least Quit, Help, and About.
- Redo all of the rooms so that they represent some part of WOU, so you are now wondering about a building, the campus, or outside locations at WOU. Have at least 10 different locations at WOU, and pictures of those locations.

Everything else is pretty much up to you. Especially: the command input method is up to you to decide. You could still have text commands (as in the picture above), or you could use a set of buttons for the commands. You decide.

Optional: challenge tasks

Here are some things you could do if you feel like it:

- Read the game specification (rooms, their exits, items) from a file instead of hard-coding it. This way, the same program can play multiple scenarios and building more complex world “maps” is easier.
- Implement a ‘Save’ command that saves the game status. Of course, you need an ‘Open’ command as well then.

Submission:

Lab #5:

For lab #5 I want to see intermediate work on this project. Turn in a RUNNING project that has at least the code refactoring done (all input and output isolated to a new GUI class), and has the Frame, menu, and textboxes created and visible. None of the GUI needs to be functional (event handlers are not required) for Lab #5, but the GUI must be shown. You should however strive to do as much as possible.

Lab #6:

You have to submit the entire BlueJ project and your report using the same method as with previous labs this term. All code must be professionally written (JavaDocs and code style & format) and will be assessed for

- correctness of all requirements
- appropriate use of language constructs
- style (commenting, indentation, etc.)
- design (consideration given to cohesion, coupling, maintainability)

Submit your entire BlueJ project (zip only, no tar, 7zip, or other file compression types) folder for this lab into the CS162, Lab2 upload link (just as was done in CS161)