

CS2110 Lecture 35

Apr. 16, 2021

- DS 9 available, due Tuesday.
- HW 9 available, due Sunday, Apr 25.
- Quiz 4 next Friday
- Two more HW and DS assignments during the final two weeks of class. My goal is for no one to want/need to take the final exam. *Work hard on the homework - that's where you'll learn the material - and for many people that will be enough for the grade they want*

Last time

- finished graphs - bfs and dfs - and discussed HW8

Today

- graphical user interfaces (GUIs). Covered in Chapter 15 of text

Graphical User Interface (GUI) Programming in Python

- tkinter is a commonly used Python layer on top of a standard GUI toolkit TCL/Tk
- GUIs built using *widgets*: basic building blocks including buttons, labels, text, entry fields, scroll bars, etc.
- Steps:
 1. Define widgets and layout
 2. Specify how to handle “events” (button presses, mouse clicks, etc.)
 3. Start GUI/“event loop”

A bit on global variables since we will use them in GUI-based programs

- Previously discussed variable scoping. In functions, local variables are created for 1) formal parameters, and 2) any names that appear on left hand side of assignment
- What if you *want* to assign to top-level variable? You can use “global”. Within a function

```
global x
```

tells Python that a local variable should *not* be created for x. Instead, uses of x within the function refer to top-level variable x
- Examples: globalVars.py
- General advice: minimize use of global variables. Functions are easier to understand if they can be understood *locally*, without having to take into account a larger context (perhaps looking through and disentangling many files/functions). *However*, when carefully used they can be very useful.
- See textbook section 15.26 for discussion of use of global variables in GUIs.” Chapter 11 of the non-interactive pdf version of the text has a section on the basics of global variables.
- For HWs 9-11 we might use several global variables to make it easy to share GUI-related and other values across multiple functions.

Getting started with tkinter

1. `import tkinter`
 - **NOTE:** module was called Tkinter in Python 2. Much web documentation still calls it Tkinter. Functions, etc.. are the same but make sure to use 'tkinter' not 'Tkinter' when importing module, calling functions, etc.
2. Before using any other tkinter functions, you must call `Tk()` to create root/main window and initialize Tkinter
 - e.g. `myMainWindow = Tk()`
3. Next, define other GUI widgets – buttons, labels, entries, and their placement (via `pack/grid` function calls)
4. Specify how to handle event (e.g. button presses), typically by defining “callback functions” that are called by the GUI system when it detects events
5. Finally, when ready to start execution of the GUI, call `mainloop()`
 - e.g. `myMainWindow.mainloop()`

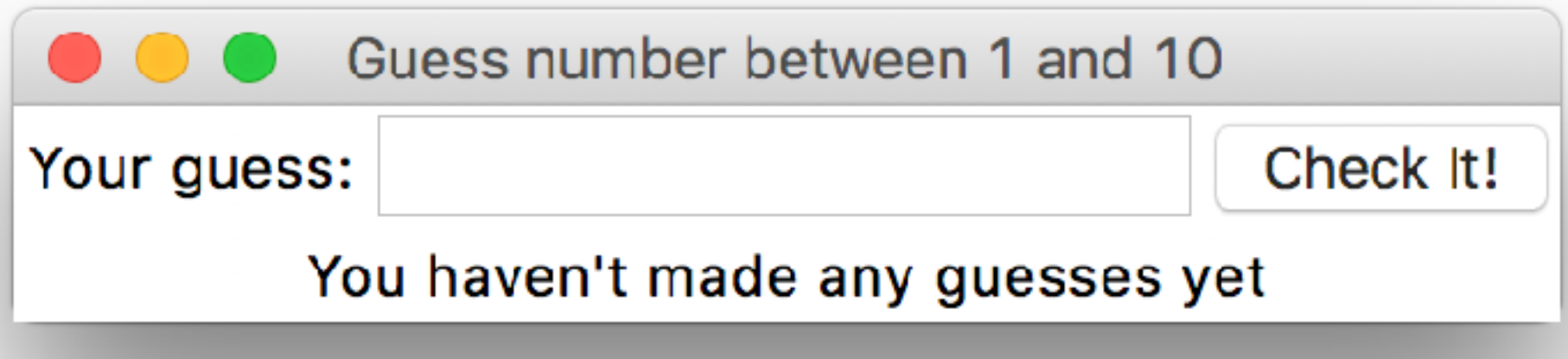
Note: `mainloop()` gives control of execution to Python. It won't return direct control to you until after you close/kill the main window. So, no commands/function calls should follow it in your code. After you call it, you only get an opportunity to change things when *events* occur – i.e. when you click the mouse on buttons, type text in entries, etc.

- tkinter is big – there are long chapters and even whole books about it. You will use Tkinter in DS9, and HW9, HW10, and HW11, but not a lot of the features.
- You just need to understand the basics of a few widget types (Label, Entry, Button, Frame) and how to respond to events like button presses.
- Make sure you understand these small examples
 - minimal.py (today)
 - minimal2.py (today)
 - simplegui1.py (next time)
 - simplegui2.py (next time)
 - simplegui2wClass.py (next time)

Links/resources for learning tkinter

- Chapter 15 of the interactive text has a LOT of info – more extensive than most of the other chapters.
- Tkinter info on the official Python site:
<https://docs.python.org/3.9/library/tkinter.html>
- This tutorial - http://www.tutorialspoint.com/python/python_gui_programming.htm - does a good job of explaining and demonstrating the basics and includes several good small examples. It seems like a good place to start.
- If you want a “real textbook chapter”, Chapter 6 of Kent Lee’s book “Python Programming Fundamentals” is pretty good. The book is electronically available to UI students through the UI library (you need to be on the campus network to access it). http://link.springer.com/chapter/10.1007%2F978-1-84996-537-8_6
- This site can also be very helpful. It’s usually the first hit when I do Google search (though I’m not sure all info there is fully up-to-date): <http://effbot.org/tkinterbook/tkinter-index.htm> *(AS OF 4/16/21, THIS SITE IS NO LONGER WORKING - NOT SURE IF IT WILL COME BACK)*
- Some of the explanations here can be helpful (e.g. explains “pack” better than many others): <http://thinkingtkinter.sourceforge.net>

DS9: simple number guessing game



1. Implement widgets and layout
2. Implement simple game behavior with just correct/not correct
3. Improve to give better feedback and allow starting a new game

ds9game.py

Modifying ds9game.py to make it a little more fun

We will improve the game in two ways:

1. We'll give better feedback to the user about the guess they've entered, displaying (in the GUI) whether the guess is too high or too low
2. When the user guesses correctly, the "Check It!" button will change behavior so that the user can press it to start a new game. (With the initial version of the game, you have to reload the code to play again.)

Next time:

- More GUI / tkinter