IP Application Development
WS 2022/23

27.02.2023

Final project – python capital cities quiz

Written by Martin Stobbelaar (12131617)

Description of the application

Goal of this application was to create a simple geography Quiz that either asks for a capital city of a given country, or the other way around. In the current version of the application, a capital city is given and the according country has to be submitted in text form. After submitting, a map opens showing the choice of the country the user chose and the correct answer (when correct obviously only the correct answer is visualized. The game lasts 10 rounds, at the end a score out of 10 will be displayed with the option to end the program or play again.

With this application, the user can get some better knowledge of the world capitals in a fun and interactive way.

Implementation

The game is programmed with python. It uses the libraries geopandas and pandas for handling the shapefile, matplotlib for creating simple maps and tkinter for the GUI. The random module is for selecting random cities.

When opening the program, two shapefiles will be loaded. These shapefile will be used as the dataset for the questions and as visualization purposes.

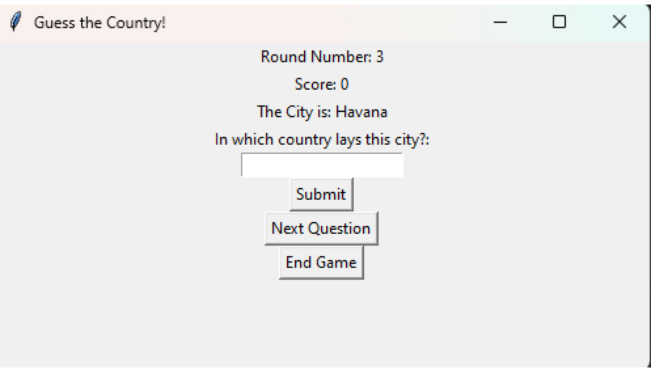
Data sources

World cities: https://hub.arcgis.com/maps/esri::world-cities/about

After downloading the shapefile, only the capital cities were extracted into a new shapefile (using ArcGIS Pro).

World countries: https://hub.arcgis.com/datasets/esri::world-countries-generalized

After loading the shapefiles with geopandas, the city data is converted to a dataframe using pandas. Using the random module, a random city will be chosen and presented in a GUI window that is created with the tkinter module. There is a input field for filling in the answer, a submit button, next question button and end game button (see figure 1).



Round Number: 3

Score: 0

The City is: Havana

In which country lays this city?:

Figure 1.: Game window

After submitting an answer, a map will be plotted (using the module matplotlib) showing a world map. The correct country will be highlighted in yellow, the given city as a blue point and if the users answer is incorrect the country in read. A example of such a plot can be seen in figure 2. Here the guess of the user is incorrect.

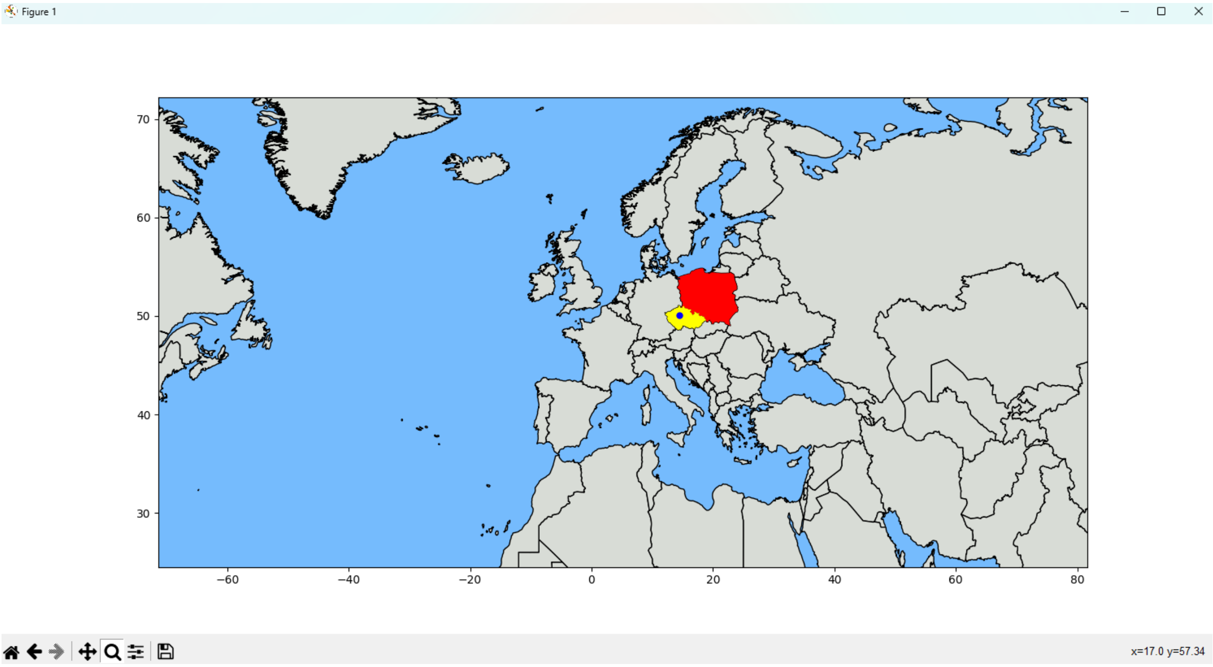
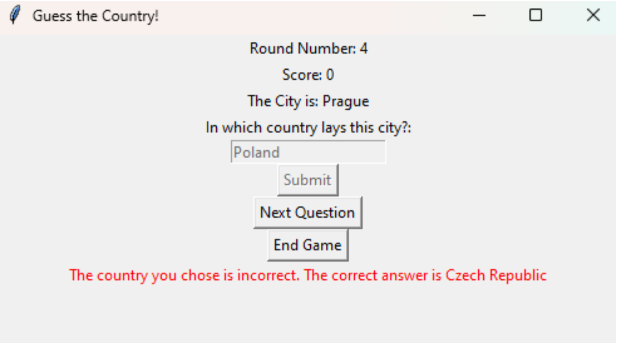


Figure 2.: Plotted map after submitting an answer (incorrect guess, zoomed in to the area)

After submitting the answer, the input field and submit button will be grayed out, the score and round number updated and a short message about the correct answer will be displayed (figure 3). The next question button closes all open windows and launches a new window with a new question.



Round Number: 4

Score: 0

The City is: Prague

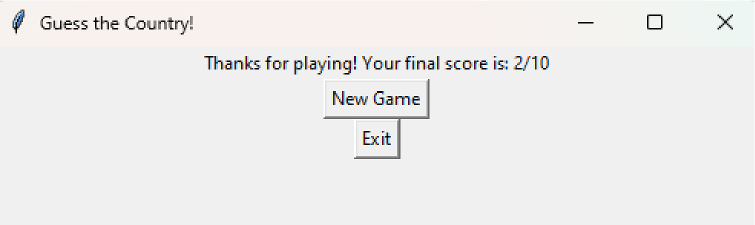
In which country lays this city?:

Poland

The country you chose is incorrect. The correct answer is Czech Republic

Figure 3.: Game window after submitting the answer

After 10 rounds (or when clicking the end game button), all windows will be closed and a window will be opened that displays the total score, an exit button and a new game button (see figure 4).



Thanks for playing! Your final score is: 2/10

Figure 4.: Game end window

All other implementations should be clear when looking at the code and inline documentation.

Challenges

1. When clicking the next question button the windows get closed as they should, when other applications are open the new question window will be opened in the background and has to be selected in the taskbar to see.

2. Dataset is not optimal as in rare cases no map is plotted. Also is the plotted map stretched in weird ways (from time to time).

3. No spelling errors are allowed. As soon as there is a small typo in the answer, it will be wrong.

Possible improvements

1. Different game modes e.g. Getting a map and try to guess the country or similar.

2. Various difficulty levels: Could be implemented by having multiple shapefiles that could be selected in the beginning depending on the users knowledge.

These and some further improvements shouldn't be too hard to program as the main code stays the same, but are time intensive.

Disclaimer

The used python modules should be imported and installed before launching the game. An alternative would be to pack the program into an executable.

The code has no QGIS or ArcGIS python module implemented. I looked into it, however the implementation was difficult.