Comp415 Project5 – getting more practice

Due: Saturday May 7th @ 11:59pm,

Objectives: to put your hard won knowledge from the semester to use one last time in a project that will appeal to you. Also now that we have explored a variety of topics in golang, it is time to have you learn just one more topic at least partly on your own to supplement what we have covered in class already.

Write one of the projects below.

For all projects: you must write at least three automated tests which run every time the you push to the github project. You don't have to test every aspect of your program but the three parts that are tested should be properly tested (use table tests or other methods to make sure you test a wide variety of good and bad test data, including edge cases)

1. Project 1) write a project which will grab an image from the camera and query an online face recognition cloud api (such as https://azure.microsoft.com/en-us/free/cognitive-services/ [this one offers a $200 30 day credit]) to identify the face in the video, At this point put a little label in a filled rectangle above the face much like the character name floating over the character's head in a video game. When you demo it to me, pretrain the face recognition on faces you have available. In your readme, explain how I should setup the training data myself to test it.

2. A game: Make a game in Ebiten. Choose between
   (a) Snake game (player controls a snake which grows when it eats. Pieces of food should appear randomly about one every 3 seconds. Each time the snake hits a piece of food, the snake grows. If the snake hits any part of itself or the edge of the window then the player loses. Each piece of food eaten adds to the score, display the score. Once every 8 seconds randomly place a snake shortener. It should remove two pieces of food worth of snake length.
   (b) A one level frogger clone, the traffic need only be four lanes wide, two in each direction. The river part can also be only four lanes wide two turtle lanes moving left and two log lanes moving right interleaved with each other.
   (c) If you want to try something else see me so we can make sure it is a similar difficulty

All games must use the Ebiten library using images included in the submission.

3. Project 3) Write a an API driven site that you host yourself (you can't use something like rapidapi to host it). Your program needs to read the "games-features.xlsx" file and store all the data into a sqlite3 database. (you can set this up as a separate mini-program that you usually

only run once). Once the data is in the database, design a simple API which will allow users to search on a game and will return the game's database entry as a json response.
Hint – check out: https://github.com/avelino/awesome-go#microsoft-excel

4. Project 4) Build a CRUD application in EbitenUI (or another crosse platform GUI if you want to learn a different one) and SQLite3, like in project2 above, read the data from games-features.xlsx into a sqlite3 database. Then write a GUI application that includes a way to search for a game record (either by gameID or name as your prefer). Once the record has been found display at least the following fields in individual labeled editable fields. Then there needs to be a way of saving. When the user selects save the data including any users edits back to the database.
   Required fields:
   (a) Name
   (b) RequiredAge
   (c) DLCCount
   (d) Metacritic score
   (e) RecommendationCount
   (f) SteamSpyOwners
   (g) SteamSpyPlayersEstimate
   (h) PlatformLinux
   (i) PlatformMac
   (j) PlatformWindows

   The three platfromXXX fields are boolean, so they can be checkboxes rather than editable text if you want

5. **5.** Project you) if you have a project idea you are dying to try instead. Propose it to me. I want this to be a substantial project that pushes you into new territory a little, but I don't want to push you too much into brand new stuff. If you have some area you are already exploring that you think you'd love to spend to weeks on, then send me an email or teams message and we can discuss it.

Make sure to use your name or github ID somewhere in the name of your project.

ReadMe: include a readme in the project that explains how to use your project

Be sure to include all of your images in your project so I can run the game.

Submission: remember to put your project on github early and push there often so that you don't lose much if your pet/child/spouse/etc should mess up your computer. Make me a collaborator to submit like in the last project.