5/16/2016

MODIFIED HANGMAN

A GAME LIKE NO OTHER…ALLEGEDLY

WHO ARE YOU AND WHY ARE YOU HERE?

- I'm Yevgeniy Liverant, a senior at Macaulay Honors College at Hunter College of the CUNY
- I'm a biochemistry major…yes…biochemistry
  - I'm a pre-medical student

WHO ARE YOU AND WHY ARE YOU HERE?

- I decided to minor in computer science because I've always been interested in everything about technology.
  - Before college I dealt with hardware, for the most part, so this was a great opportunity to learn how to code
    - I have gained experience in Python, C++, Oracle SQL, and SQLite

WHAT ARE YOU DOING?

- In classic Hangman, one person thinks of a word that she wants the player to guess, draws the appropriate number of spaces, one for each letter, and allows the player to begin guessing
- For each incorrect letter guessed, a body part is drawn on a hangman stand
- If the player completes the word before the stick figure is complete she wins
- Otherwise, she loses

WHAT ARE YOU DOING?

- This project is a modified version of the game Hangman.
  - At the beginning of the game, the player is asked to input a word, which will be referred to as the "search word" moving forward.
  - Next, all of the words that have the search word in their definitions will be found,
    - One of these words will be randomly selected to be the "game word"

DID YOU USE A DATABASE?

- In order to realize my vision for this game, the first thing I needed was a searchable dictionary.
- I found a MySQL English Dictionary Database, but it was incompatible with Sqlite 3
  - I downloaded the .txt version and created the database myself using:
    - A C++ script I wrote (for .txt formatting)
    - Termsql for Sqlite

FLASHBACK TIME!

GOALS

- Set-up an intuitive graphical user interface (GUI) that allows the user to play the game
- Interact with an English dictionary database to feed words and definitions for the game. Also to be used to search for eligible words given the user input
- Ensure that the game does justice to the original *Hangman* by containing all the components of the original game (with the additions that were mentioned)

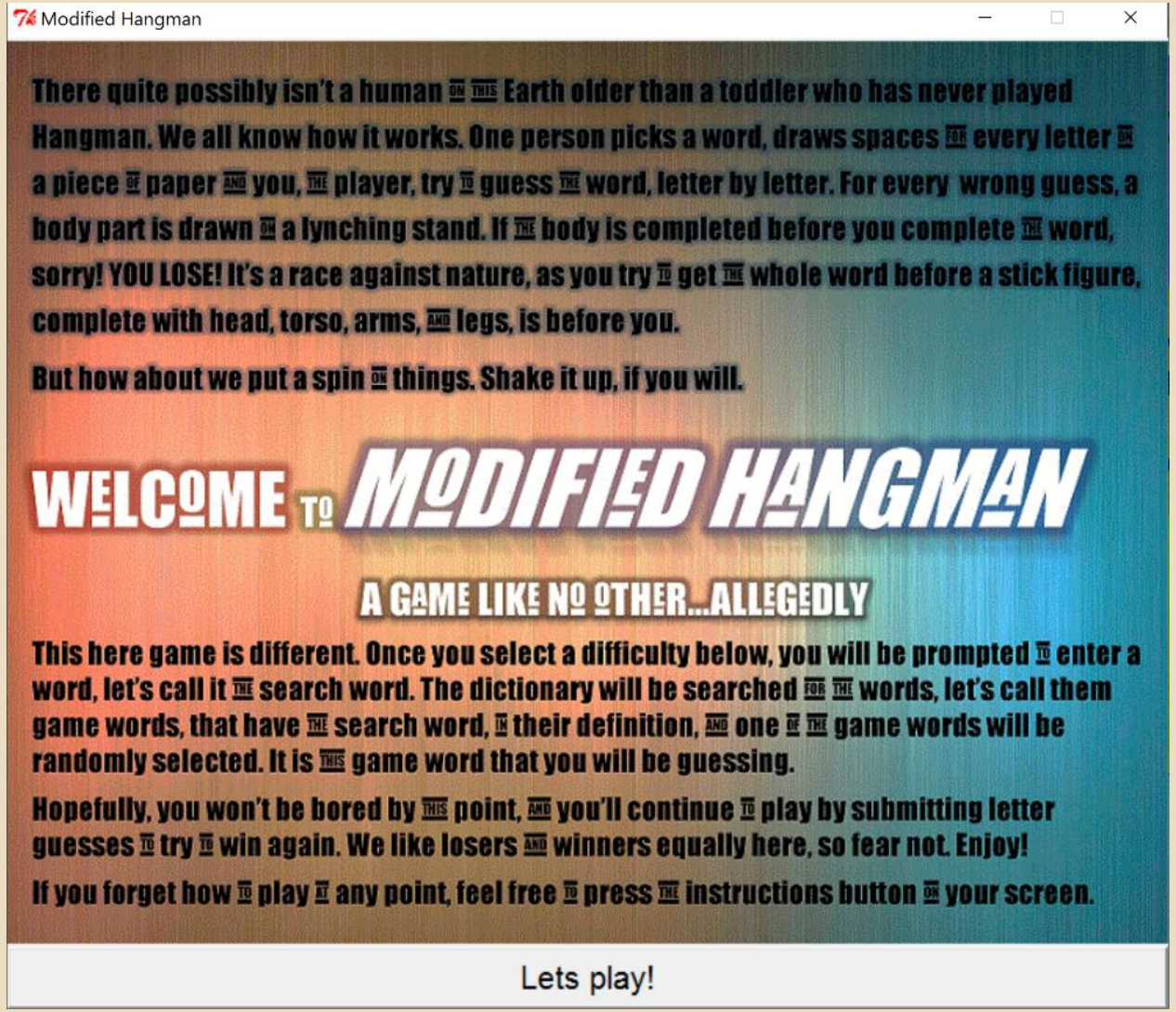
GOALS

- Display the definition of the game word as a clue, but only if the player wants it
- Make addition of hangman body parts seamless
- Provide two levels of difficulty, easy and hard, which give the player a different number of guesses

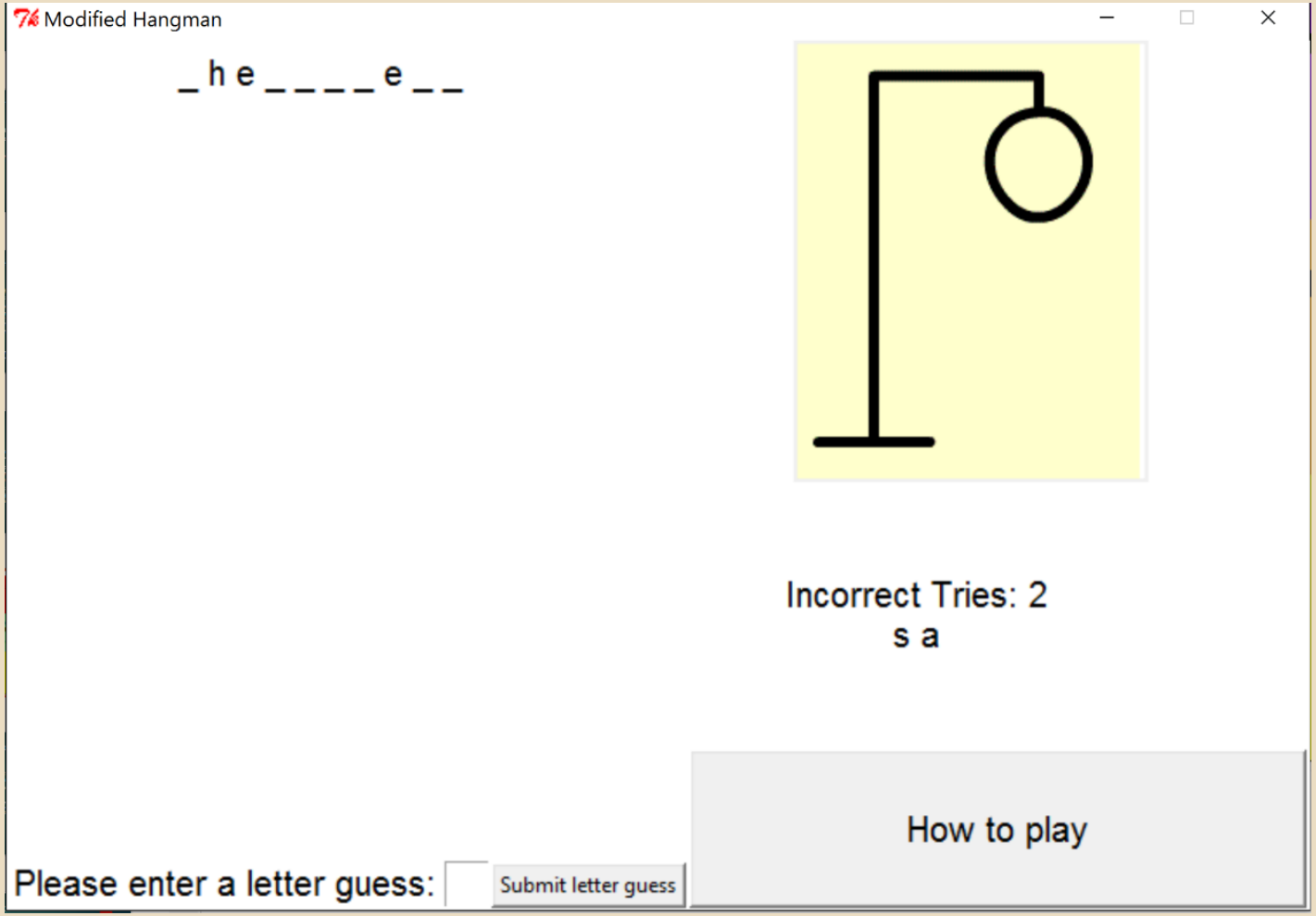
GRAPHICAL USER INTERFACE (GUI) - AT THE TIME OF THE LAST PRESENTATION

FLASH FORWARD!

GRAPHICAL USER INTERFACE (GUI) - NOW



GRAPHICAL USER INTERFACE (GUI) - NOW



GRAPHICAL USER INTERFACE (GUI) - NOW

Please enjoy this demonstration of *Modified Hangman*

UNDER THE HOOD

- Python 2.7
  - Tkinter
    - Module used for creating the GUI
- SQLite 3
  - Used to manipulate and interact with he dictionary database
- C++ 11
  - Used for manipulation of dictionary .txt file
  - Used instead of Python because of better familiarity with file I/O in C++

UNDER THE HOOD

- Python 2.7
  - Tkinter
    - Module used for creating the GUI
    - I created many widgets - Windows, Frames, Buttons, Labels (text and image) and Canvases
      - Type of widget was carefully selected to fit the desired use while minimizing syntactical complexity

UNDER THE HOOD

- SQLite 3
  - Used to manipulate and interact with he dictionary database
    - Termsql, a plugin for SQLite 3, used to convert my .txt file to a database
  - I used the sqlite3 module in Python to perform the dictionary search and selection of the game word

UNDER THE HOOD

- C++ 11
  - Used for manipulation of dictionary .txt file
  - Used instead of Python because of better familiarity with file I/O in C++
    - Having just learned file input/output in C++ in the weeks leading up to when I created my database, I was more familiar with how to accomplish what I needed using a C++ script

UNDER THE HOOD – ISSUES AND TOOLS

- When I found the English Dicitonary database on Sourceforge, with Felix's help, I was thrilled
- Unfortunately it turned out that this database was written for MySQL and would not be recognized by SQLite 3
- I spent hours trying to get it to work, to no avail

UNDER THE HOOD – ISSUES AND TOOLS

- I then downloaded this same database as a .txt file, available from the same Sourceforge page
- I again spent hours, over the course of multiple days, pondering over how I would convert this file to a SQLite 3 database
- In doing my research I happened to stumble upon Termsql

UNDER THE HOOD – TERMSQL

- A project by Tobimensch available on GitHub
- SQLite 3 tool for conversion of plain text to tables in a SQL database
- This would take care of the conversion, but there was another issue
- Termsql allows for no more than one delimiter to be specified

UNDER THE HOOD – MY DATABASE

- The text file I obtained is structured as follows:

  Word (part of speech) Definition.

- My plan was to use '(', ')', and '.' as delimiters, but Termsql only allows for one delimiter to be used
- Enter C++….

UNDER THE HOOD – MY DATABASE

- In order to deal with this limitation, I decided to write a C++ script to re-write the entire dictionary text file to have a '+' between each area. Each entry would then be:

  Word+(part of speech)+Definition.+

- Several hours later, my script was almost working, albeit slowly (the input file is 14.7 MB after all), but the '+' at the end of each entry was missing and I couldn't find a way to fix it.
- And then I re-wrote the whole script….

UNDER THE HOOD – MY DATABASE

- After all this work, it hit me. My approach was wrong.
- I decided to add the '+' signs in the right places in the input file, instead of creating a new file
  - Allowed me to avoid end of line issues I was having
- Another several hours later (déjà vu, no?), my script was working as I had wanted it to work

UNDER THE HOOD – TERMSQL

- And so I was now able to use Termsql to create my SQLite 3-compatible database
- Now, the real fun would begin

UNDER THE HOOD – ISSUES

- Preventing the player from submitting a blank search word
  - Resulted in crash of program when it happened
  - Fixed by showing a warning when the player tried to do this

UNDER THE HOOD – ISSUES

- Preventing the player from submitting more than one character as a letter guess
  - Two Classes
    - ValidatingEntry – allows for constriction of input possibilities
    - MaxLengthEntry – for specifically limiting entry length
      - Sub-class of ValidatingEntry

UNDER THE HOOD – "ISSUES"

- The intricacies of Python as a language gave me major issues, especially early on
  - Strings are immutable
  - Ints are immutable
  - All objects are passed by reference
  - Forced spacing
  - Inability to forward declare functions

UNDER THE HOOD – "ISSUES"

- The intricacies of Python as a language gave me major issues, especially early on
  - Strings are immutable
  - Ints are immutable
- In order to overcome these issues, I used lists of string and int objects in code not directly related to GUI elements, as elements of a list, no matter the data type, can be updated

UNDER THE HOOD – "ISSUES"

- The intricacies of Python as a language gave me major issues, especially early on
  - All objects are passed by reference
    - That is, the address to the object is passed in and out of functions, rather than the value or values contained in the functions
    - If you're a C/C++ programmer, the option to pass objects by reference is a godsend
    - Because Python doesn't have pass by value, pass by reference is the default, which makes returning multiple objects in a function more difficult

UNDER THE HOOD – "ISSUES"

- The intricacies of Python as a language gave me major issues, especially early on
  - Forced spacing
  - Inability to forward declare functions
- These two "issues" have more to do with style than anything else, but these aspects of Python gave me functionality problems as well.

NEW GOALS - ISSUES

- Game Over/Try Again functionality became a stretch goal just out of my desire to really make this game feel complete. However, resetting the GUI was more complicated than I expected
  - I tried to make the program restart, to no avail
    - I couldn't find a universal (operating system-independent) way to accomplish this, so I decided against it
  - I could accomplish the resetting of all of the graphical user interface elements "automatically"
    - I ended up having to reset each element manually

NEW GOALS - ISSUES

- Having a welcome screen was another idea that I had that wasn't originally going to be part of my game, but why not?
  - Adding background images to Tkinter GUI is an adventure
    - Canvas functionality (used for my images of the Hangman) was syntactically complicated for this purpose
    - I overcame the issue using PhotoImage and Labels

NEW GOALS - ISSUES

- Using PhotoImage and Labels for the Welcome Screen
  - This method was not without complications
  - Python garbage collection resulted in my image being deleted before it even displayed, because it was too large
    - Fixed by a syntactical workaround

IT'S GETTING THERE…!

- There still remain certain issues that need to be worked out
  - Complete reset of graphical user interface upon choosing to try again
  - Disallowing words within longer words from qualifying when the dictionary search is performed
- Difficulty options are a work-in-progress

WHAT DID YOU LEARN?

- Stop thinking in C++
- If there's a will, there's a way
- Cleaner =/= Better
- Be thankful for things the language takes care of for you

THANKS FOR LISTENTING!

QUESTIONS?

BONUS!!! – TOPLEVEL IS AWESOME!

- Tkinter functionality that allows for additional windows to be created over the "master" window
  - Same functionality as the master windows
  - Useful for:
    - Messages
    - Warnings
    - Additional options
    - Etc.

A BIG THANK YOU TO:

Tobimensch for Termsql –

https://github.com/tobimensch/termsql

Fredrik Lundh for ValidatingEntry and MaxLengthEntry Classes –

http://effbot.org/zone/tkinter-entry-validate.htm

Espeon200 for Hangman Images –

http://media.photobucket.com/user/Espeon200/media/hangman/stage1.png.html?filters[term]=hangman&filters[primary]=images&filters[secondary]=videos&sort=1&o=11

PlanWallpaper for Welcome screen Background image –

http://www.planwallpaper.com/static/images/518079-background-hd.jpg