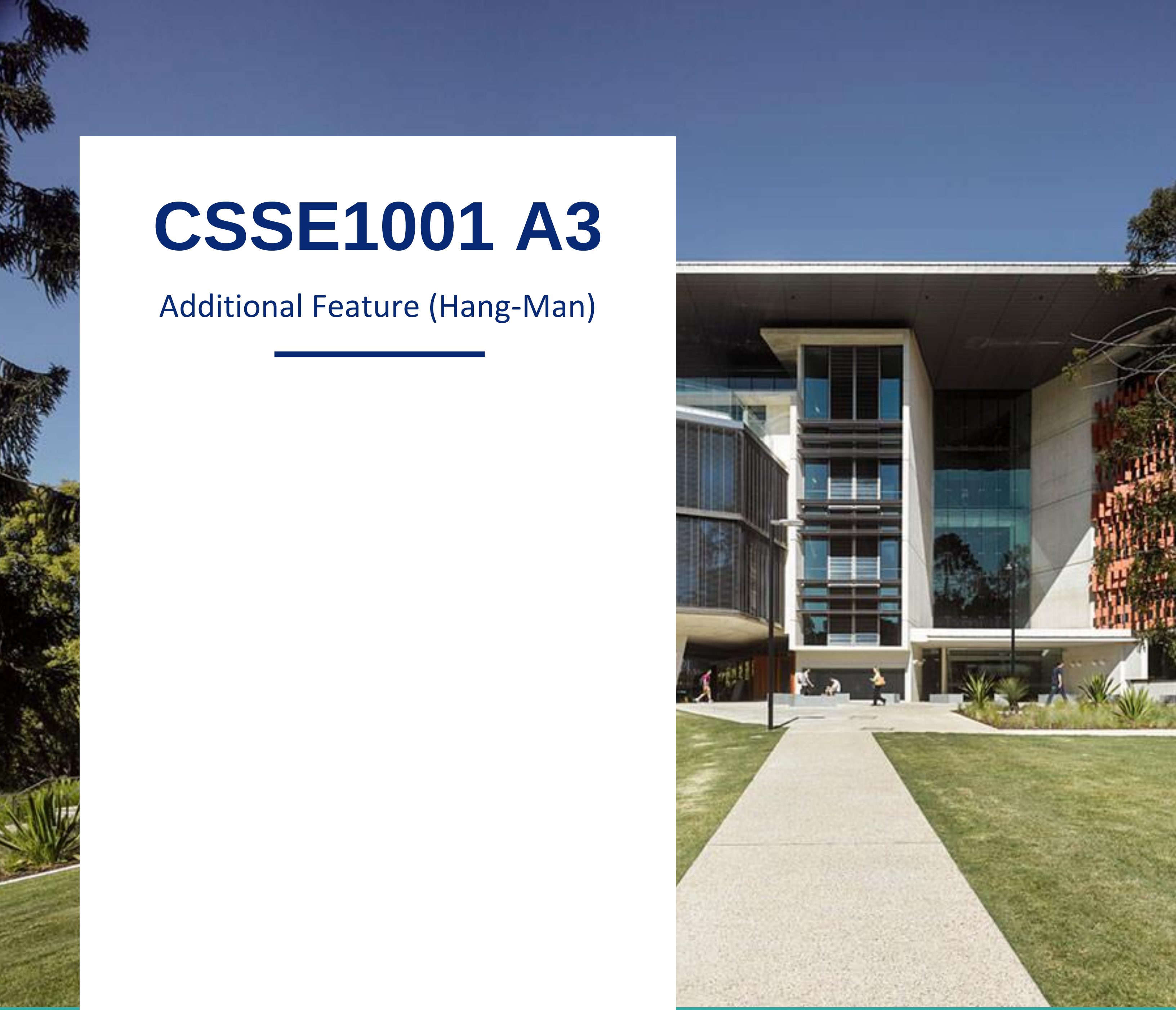


CSSE1001 A3

Additional Feature (Hang-Man)

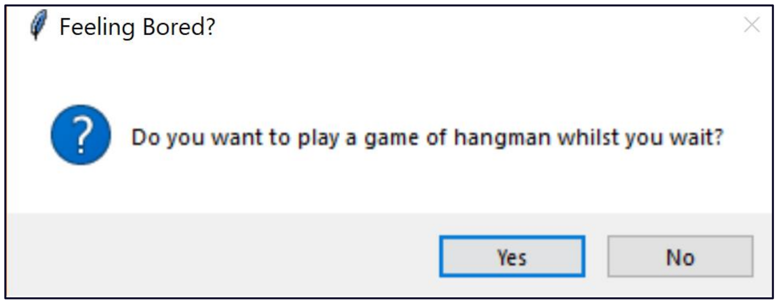
OCTOBER 21

Semester 2, 2018
Authored by: Blake Rowden

Hang-man

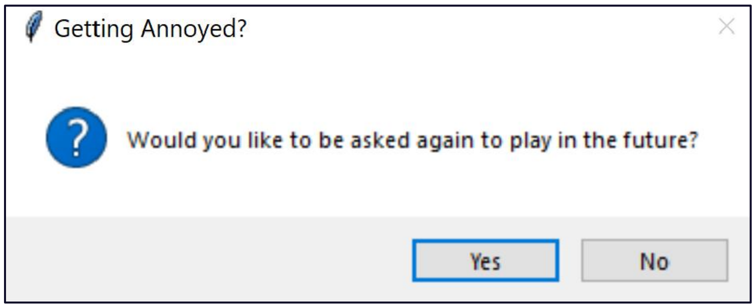
The user manual and description.

To get playing right away all a student needs to do is simply join a queue and a prompt will appear asking if you would like to play a game of hangman whilst you wait. Press yes and the game will begin, press no and the application will return you back to the queue application.


Feeling Bored?

Do you want to play a game of hangman whilst you wait?

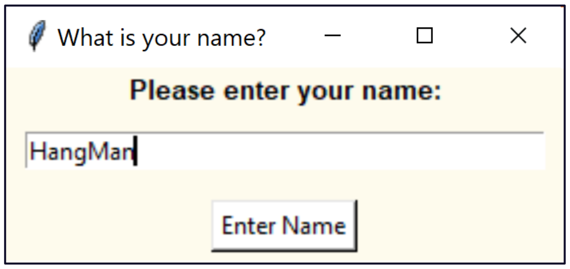
After being asked if the user would like to play, another prompt will arise. This prompt asks the user if they would like to stop receiving messages about playing a game every-time they join a queue. If they select no to being asked in the future, the game prompt will cease to show when the student joins a queue.


Getting Annoyed?

Would you like to be asked again to play in the future?

"We interrupt this program to annoy you and make things generally more irritating"

An Easter-egg way of entering the game is to enter in the words "HangMan", "hangman" or "PlayGame" into the name box when the queue application asks for a name. This is the only way to enter the game is the user selects no to being asked to play.


What is your name?

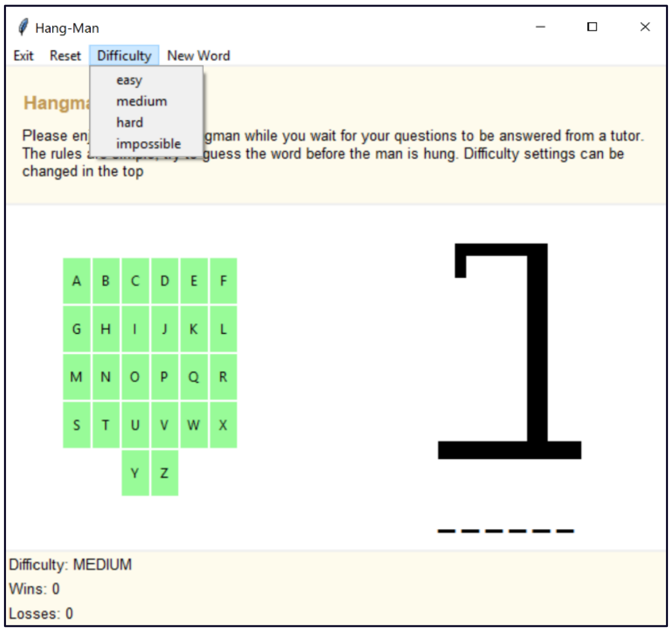
Please enter your name:

HangMan

"Remember it's not a bug, it's a "feature"."

Once in the game window a word will be selected from random based on a difficulty setting. The default is Normal.

The user may change this difficulty setting at any time using the menu at the top. Easy to Hard is based of length of words and Impossible is a special list of very hard words with unusual letter structure.

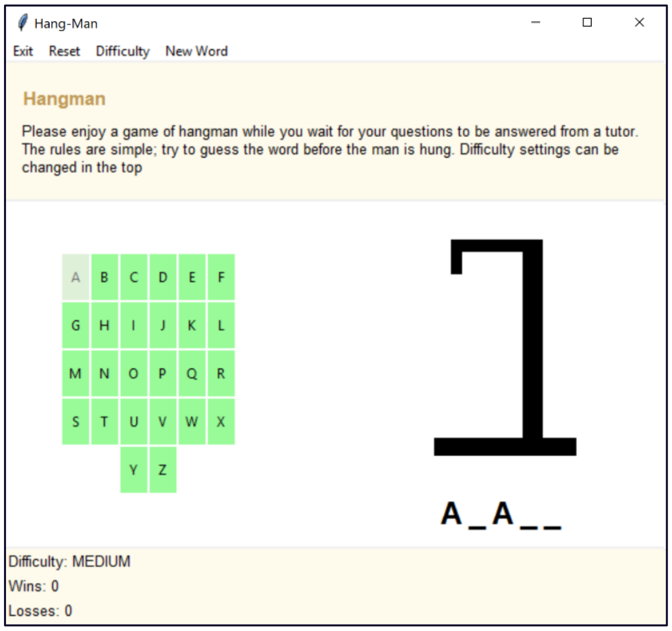


Hangm

Please en gman while you wait for your questions to be answered from a tutor. The rules guess the word before the man is hung. Difficulty settings can be changed in the top

Difficulty: MEDIUM
Wins: 0
Losses: 0

The rules of the game are the same as the rules of standard hangman. The computer, after selecting a word at random, will reveal the number of letters in the word represented by an underscore. The user will continue to take guesses of possible letters in the word. If they guess a correct letter its locations will be revealed on the board.

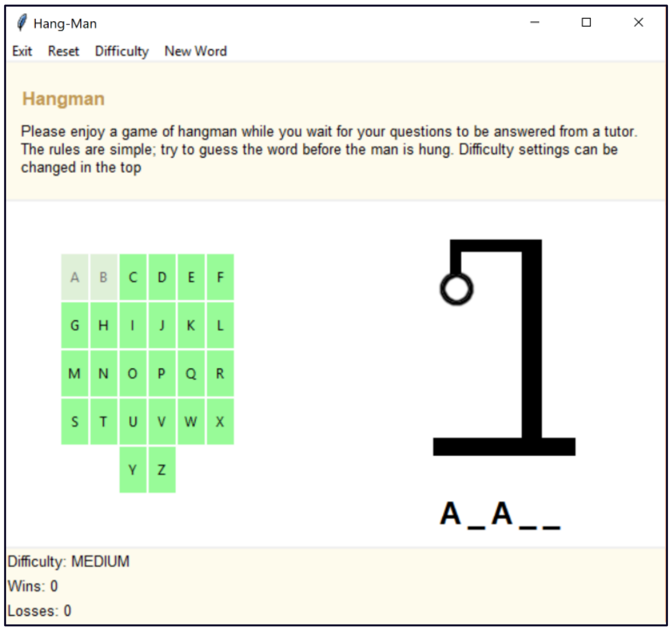


Hangman

Please enjoy a game of hangman while you wait for your questions to be answered from a tutor. The rules are simple; try to guess the word before the man is hung. Difficulty settings can be changed in the top

Difficulty: MEDIUM
Wins: 0
Losses: 0

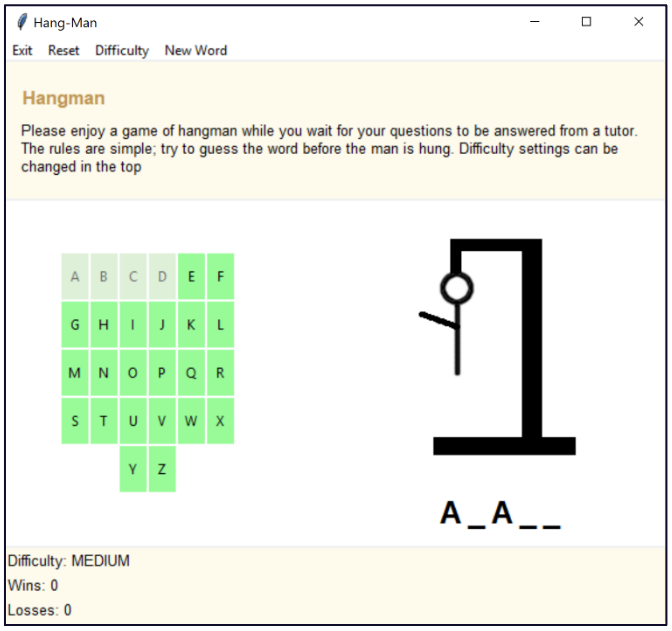
However, if the user guesses an incorrect letter the hangman sprite will advance to the next stage.



Hangman

Please enjoy a game of hangman while you wait for your questions to be answered from a tutor. The rules are simple; try to guess the word before the man is hung. Difficulty settings can be changed in the top

Difficulty: MEDIUM
Wins: 0
Losses: 0

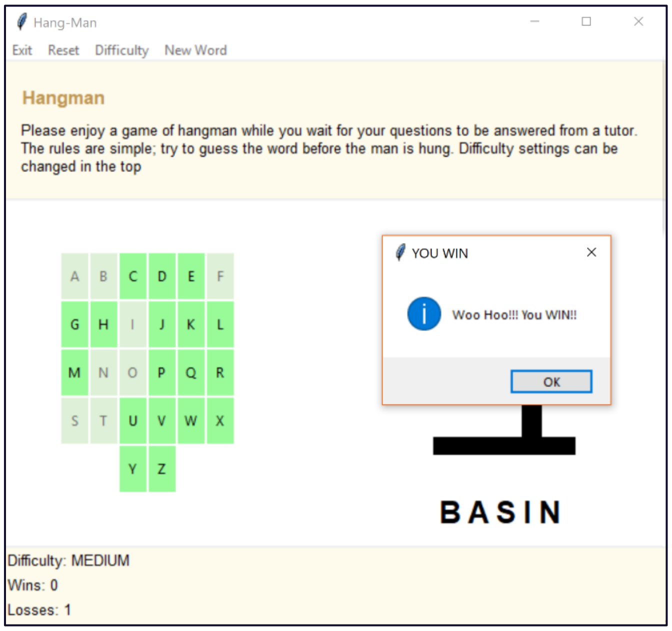


Hangman

Please enjoy a game of hangman while you wait for your questions to be answered from a tutor. The rules are simple; try to guess the word before the man is hung. Difficulty settings can be changed in the top

Difficulty: MEDIUM
Wins: 0
Losses: 0

There are two possible outcomes for every game. Either the user wins buy guessing every letter before the man is hung.



Hangman

Please enjoy a game of hangman while you wait for your questions to be answered from a tutor. The rules are simple; try to guess the word before the man is hung. Difficulty settings can be changed in the top

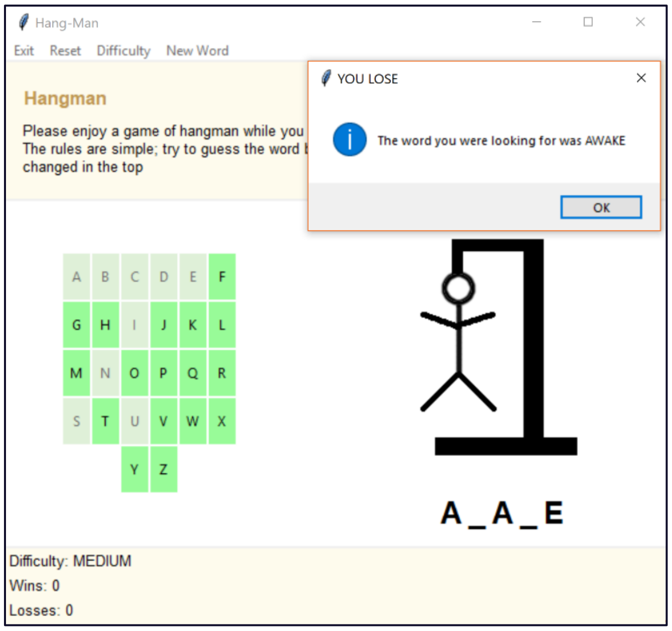
YOU WIN

Woo Hoo!!! You WIN!!

BASIN

Difficulty: MEDIUM
Wins: 0
Losses: 1

Or the user loses by making too many incorrect guesses. The word will be revealed to the player.



Hangman

Please enjoy a game of hangman while you
The rules are simple; try to guess the word
changed in the top

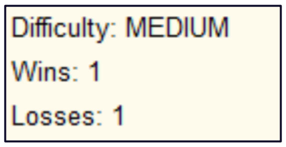
YOU LOSE

The word you were looking for was AWAKE

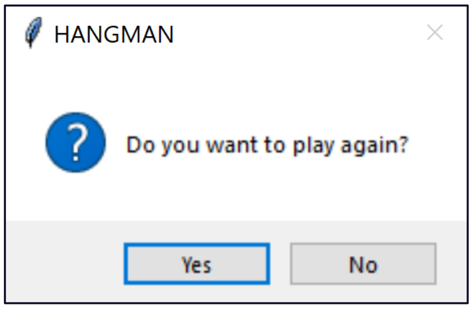
A _ A _ E

Difficulty: MEDIUM
Wins: 0
Losses: 0

Both outcomes will be added to the game totals at the bottom of the screen.


Difficulty: MEDIUM
Wins: 1
Losses: 1

At the end of every-game the application will ask the user if they want to play another game. Yes, will generate a new word and No, will return the user back to the queue application.


HANGMAN

Do you want to play again?

There is an interactive menu at the top of the screen that will allow the user to; Exit the game, generate a new word, change the difficulty or do a hard reset clearing all data from the game (total wins and losses).

Approach to writing the code

My goal was to add a game into the queue application in a way that was seamless. I didn't want a simple button placed into the application that said GAME HERE. I decided to achieve this by making a prompt every-time a student joined a queue.

To make the top window I watched some tutorials on YouTube explaining multi-window tkinter applications and implemented some version of that code to create a skeleton class that I could build my game on.

After having to test some queue functionality. I began to realize that the constant window prompts are extremely exasperating, so I made a second prompt asking the user if they would like to stop receiving prompts.

When building the hangman application, I first built it using print statements only without tkinter integration to test that the idea would work. The first hurdle I ran into was obtaining a large list of words and randomly generating one. I began by simply adding in all English words from the words.txt file used in CSSE1001 a2 support files, however the words generated were mostly not every-day English. I went for a search on git and found a list someone else used for a hangman game that contained acceptable words.

The next issue I faced was designing a difficulty setting I decided to go with word length i.e. short to very long being relative to difficulty. To select a word, I used recursion to generate a word within a given length from a list of imported words. After testing I found that some short words were also very difficult, so I filtered the hardest words I could find and added them at the end of the list. I made a new difficulty setting called impossible and setting the random generation index to the words added at the end of the list.

Next issue I found was that when the user selected a letter it would only reveal the first letter. If there were multiple A's or E's in a word only one would show. So, after searching stack overflow for similar issues I eventually came across the idea of using a list comprehension which worked perfectly.

After creating a working game in console, I converted it to tkinter. The main issue I had was with the virtual keyboard. Getting each key to function with a slightly different command and fit into a different location whilst not repeating so much code was a struggle. I ended up creating a function that created a button given some inputs, then running this function 26 times placing the results into a dictionary with the key being the respective letter. The creation of the dictionary proved to be extremely useful when deactivated each button when pressed.

Finally, when developing the look of the game I took inspiration from the layout of the queue application. This helped with my seamless feel. I used the header class created for the queue app and used the same color scheme.

After many test plays and minor bug fixes it appears to be functioning perfectly. On reflection there are many things I could have done better with more careful planning and took a more efficient approach, but I am happy with the current outcome and have learnt a lot.