**Hangman Project**

John Cason

Bellarmine University

CS150-01

Professor Sarkar

12-13-2023

**Abstract:**

With the world continuing to push more and more toward convenience and technology, I decided to learn possibly the most valuable tool on campus, coding. In the CS150 class we have learned many different topics and coding strategies through Python. Through these things I have learned I have been able to develop a hangman game testing my knowledge and understanding of the coding I have been taught. This paper will provide an overview of my program's purpose, the code used, its key features, the challenges and results of the code.

**Introduction:**

The world is moving towards technology and with that technology has become a sport in eGaming. This idea pushed me to create one of my all-time favorite games in hangman. Creating a game like this is not only rewarding, but also is a great way to challenge your knowledge of coding. To create a game, you cannot just code, but also have to design the game logic and displays used in it. When creating any game, you must always have a solid understanding of the building blocks of coding to create simple, useful, and effective code.

**Methodology:**

The Hangman program I have created uses a variety of programming elements to ensure optimal functionality. By using the versatility of while loops, for loops, and if functions, the program can navigate through various inputs of letters and words, providing the correct response to user inputs as they are guessed. Lists and strings are also used to efficiently organize and manage inputs, while the append function facilitates the inputs and enters them into strings and lists that will display during the game. Additionally, random is used in the code so you can get a new random word every time you play. There are also functions and tools used like isalpha, enumerate, and .join to enhancing the program's ability to handle incorrect inputs and correctly return displays after guesses.

**Design and Implementation:**

I attempted to take a simple approach to creating hangman and tried to only use functions and things we learned this year. The hangman game is made up of three functions. The first function is the get\_word function (before creating this function I imported random). In this function I created a list of 20 words that had to do with Python and coding. I used this list and the random function to return a random word from the list that the player can guess. After this I created the display function which I used to create a list of all the images needed for hangman. I then had the function return the image that correlated with each part of the game by setting equal to guess available.

Finally, I created the play function. This function is where the game was created. I started this function off by giving welcoming messages and the category of the word the user will be guessing. I then created a group of lists that I would need to help manage the data. I also created the amount of attempts the player would have. After this I created the looping part of the game using a while loop and if statements within it to check if the guessed letters and words were correct. If they were not, then an attempt would be taken away and the correlating image would display. This would continue until the player either guessed the word or ran out of attempts.

**Challenges and Solutions**

The biggest challenge was the play function. I really struggled with getting all of the letters and words to append the right way and spent many hours researching to figure it out. It was hard to start coding the program because I decided to start with the while loop and as I realized that I would need something I would go and create it. I think this is the best way to start any coding project. It is really hard to create stuff you need before you realize you need it. For example, as I started creating the while loop, I realized I needed multiple lists. One list for the guessed letters and one for guessed words. As I continued through the coding, I could feel it just start to flow as I continued to gain a better understanding. I also experienced some other challenges with a lot of trial-and-error outputs. I also had to correctly use the continue and breaks to make sure the code wouldn’t continue after the correct word was guessed.

**Results**

The first result and finding I had to think about was the player experience. This is when I started to focus on the little things like printing welcome to hangman at the beginning of the game and having all the guessed letters and the category print out as well. The second result was making sure the game ran smoothly and accurately. I had to make sure the player could guess both letters and words and the number of attempts was accurate. I also made sure the had the hangman image pop up with all of stages of the game.

**Discussion**

This project was extremely helpful to giving a more Indepth knowledge about the application of while loops and if statements. I gained a better knowledge of all coding through this and was able to better understand the basic and core theme of coding and appreciate the challenges. For the project itself, the players will be able to enjoy a fun game and pass time. I think players will be able to appreciate the quality of the game that has been presented before them. This program gave me many insights into the challenges coders run into every day and gave me an appreciation for their job.

**Conclusion**

In conclusion, this project gave me a deeper understanding of all the elements and tools to python we have learned this year. By making programs from scratch, I get a clearer sense of how useful these functions are. Through this hangman program I also gained a deeper understanding of lists, while loops, and if conditions. The knowledge gained from an individual coding project cannot be replicated in class or in a group. This challenging process has helped grow in my coding.

**References**

* Module 1 – Introduction to Python
* Module 2 – Lists and Tuples
* Module 3 – If Statements
* Module 5 –While Loop