YZV104 TERM PROJECT: LEXİ-Q

A Word Guessing Game

SUMMARY

Lexi-Q is a word guessing game designed to enhance players' vocabulary skills while providing an enjoyable gaming experience. The game offers three distinct modes, where players attempt to guess the secret word by entering various words and receiving feedback on the accuracy of the guessed letters and their positions. In the event that players are unable to guess the word within the maximum allowed attempts, they have the opportunity to learn the meaning of the secret word. Additionally, if players accumulate enough points in the mini snake game, they can unlock the chance to play again. That adds an extra layer of excitement and motivation for players to improve their performance. The game utilizes a modular approach, implementing separate functions and a class to handle different aspects of the game. This report provides a summary of the Lexical Quest, including its purpose, logic, method, findings, conclusions and more.

Abdulkadir Külçe -- 150210322
Istanbul Technical University
29.05.2023

Table of Contents

Introduction

The Lexi-Q is an interactive program designed to provide an entertaining and challenging experience for users. In this game, players have the opportunity to test their word-guessing skills by attempting to uncover a secret word within a limited number of attempts. The purpose of this report is to provide an overview of the Lexi-Q, its development process, and the key findings and conclusions derived from its implementation.

The scope of the Lexi-Q revolves around the concept of word guessing and feedback. Players are presented with a secret word of a certain length, and they must enter words to guess the letters and their correct positions in the secret word. Through a systematic approach of trial and error, players aim to unravel the mystery word within a predefined number of attempts.

The development of the Lexi-Q involved the utilization of various programming techniques and methodologies. The program incorporates a random word selection process, user input validation, and a scoring system to track the performance of players. Additionally, the game provides feedback on the correctness of guessed letters and their positions, enhancing the player's understanding of the secret word.

Throughout the development process, several aspects were considered to ensure an engaging and user-friendly experience. These include the selection of an appropriate word length, the determination of the allowed maximum number of attempts, and the implementation of a scoring system that incentivizes accurate guesses. The result is a captivating game that challenges players' word recognition skills and strategic thinking.

In conclusion, the Lexi-Q provides an enjoyable and intellectually stimulating experience for players. Its development involved careful consideration of various factors to create a balanced and engaging gameplay environment. By offering players the opportunity to test their word-guessing abilities and providing feedback on their progress, the game aims to entertain and challenge users while enhancing their cognitive skills. The subsequent sections of this report will delve deeper into the specific methods employed, the discussion of the game's features and mechanics, and the conclusions drawn from its implementation.

Methods

The Lexi-Q is implemented using the following methods:

LexicalQuest(): This is the main function that starts the Lexi-Q game. It prompts the user to choose a game mode (5, 6, or 7 lettered mode) and initializes the game. It calls the **Lexi-Q** class and sets the game mode. Then, it selects a random word from the word set file corresponding to the chosen game mode. The function continues the game loop until the maximum number of attempts is reached or the word is correctly guessed.

graph_of_progression(): This function generates a graph of the player's score progression throughout the game. It uses the **tkinter** library to create a window and **matplotlib** to plot the data. The x-axis represents the number of tries, and the y-axis represents the closeness score. The function retrieves the updated scores from the score list and plots the data. The tkinter event loop is started to display the graph.

take_word_list(number_of_letter): This function is responsible for retrieving and storing word lists from a website. It takes the number of letters as input and constructs a URL to visit the corresponding web page. It sends a GET request to the URL and uses **BeautifulSoup** to parse the HTML content of the page. It then extracts the words from the page and stores them in a list. The word list is sorted alphabetically, and the words are written to a text file with the naming convention "ordered_{number_of_letter}_lettered.txt".

Lexi-Q class: This class represents the Lexi-Q game. It has the following methods:

- **__init__(self, word)**: The constructor initializes the game attributes. It takes a secret word as input and sets the remaining attempts, and other necessary variables.
- **game_mode(self, mode)**: This method sets the game mode based on the user's input. It determines the word length and other game parameters according to the chosen mode.
- **which_word_set(self, filename)**: This method selects a random word from the word set file corresponding to the game mode. It reads the words from the file, selects a random word, and returns it as **self.secret**.
- **attempt(self, guess)**: This method handles a player's guess. It takes the player's guess as input and updates the game state accordingly. It appends the guess to the attempts list and other variables changes according to this.
- **guesscheck(self, guessing: str):** This methods take players' guess and *turns a list of letters and the bool value of whether that letter in the word and the position of it is correct that are in secretword*
- **colored_result(self,result):**This method converts entered word into colored_word by looking the letters one by one and resigning a proper color value to each by looking at the boolean values coming from **guesscheck(self, guessing: str)** function. And also this method calculates tempscore by assigning some predefined numbers for green ,yellow and white letters and multiplying them with remaining attempts. Finally it adds tempscore to score list and returns colored_word.
- **show_results(self)**: This method displays the current game state to the player. It shows the attempts made on terminal screen with color by looking at the words coming from **colored_result(self,result)** function.
- **is_solved(self)**: This property returns True if we still can make a prediction and last 'self.secret' is the secret word. It is checked after each guess and if it returns True the game stops.
- **remaining_attepts(self)**: This property updates the number of remaining attempts.
- **can_attempt(self)**: This property checks whether we can make a new prediction or not. It makes it by looking at the datas coming from the two function above this function. And decides we still can play the game after each time we guess.
- **meaning(self, word)**: This method asks the player if they want to learn the meaning of the secret word. If the player chooses to learn the meaning, it searchs the meaning of the Word on default browser by using **webbrowser** module.

Also, there is another game in my main project, it is under 'MySnakeGame' folder. Various methods have been used to create that game too. I have designed all visual things in that game using paint. I designed minigame using **turtle** that allows users to create and control turtle objects on a graphical window, it provides a simple and intuitive way to create drawings, animations, and games. Also, to add some sound effects in the game I have used '**playsound'** library in my minigame.

Overall, the Lexi-Q implementation follows a modular approach, with separate functions and a class to handle different aspects of the game. It includes features like scoring, graphical representation of score progression, and the ability to retrieve word lists dynamically.

Discussion

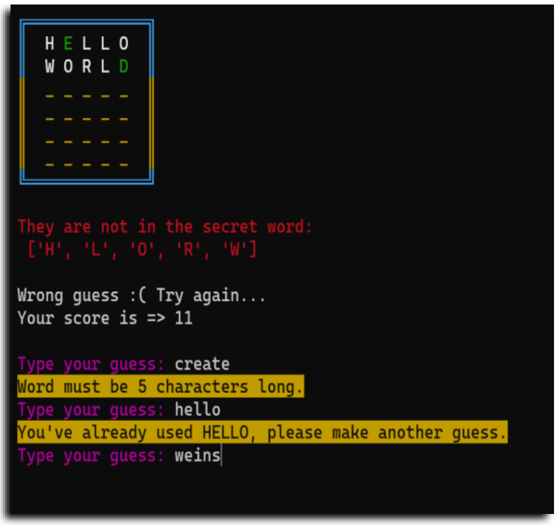
The Lexi-Q offers a captivating and challenging experience for players, encouraging them to use their word recognition and deductive reasoning skills. This section of the report will discuss the key aspects of the game, including the game mode, word selection process, feedback mechanism, and scoring system.

One of the fundamental features of the Lexi-Q is the game mode, which allows players to choose the difficulty level by selecting the number of letters in the secret word. The game currently supports three modes: 5-letter, 6-letter, and 7-letter words. This customization option provides players with varying levels of challenge and accommodates different skill levels and preferences.

Also, to enhance the Lexi-Q and provide a wide range of words for players to guess, an external source is utilized to generate a comprehensive word list. The implementation of web scraping techniques allows the program to fetch words of a specific length from the website "**wordfind.com**" and store them in an ordered manner. The "take_word_list" function utilizes the Python "requests" library to send a GET request to the desired URL, which is dynamically generated based on the number of letters specified. The response from the website is then parsed using the "BeautifulSoup" library, specifically the "html.parser" module. This enables the extraction of relevant information from the HTML content of the website. By inspecting the HTML structure of the webpage, the function identifies all "li" elements with a class attribute value of "dl". These elements contain the desired words within the "a" tags. The text content of each "a" tag is extracted and added to the "word_list" variable.

To ensure convenient usage and improved user experience, the word list is sorted alphabetically. This ordering allows players to easily locate and guess words during gameplay. Finally, the function saves the generated word list into a text file named "ordered_{number_of_letter}_lettered.txt", where "{number_of_letter}" represents the length of the words being fetched. Each word is written on a separate line in the text file, making it easily readable and accessible for future use. This process of fetching and ordering words from an external source ensures that the Lexi-Q can offer a diverse and extensive vocabulary for players to engage with. The integration of web scraping techniques adds a dynamic and up-to-date element to the game, as the word list can be refreshed periodically to include new words.

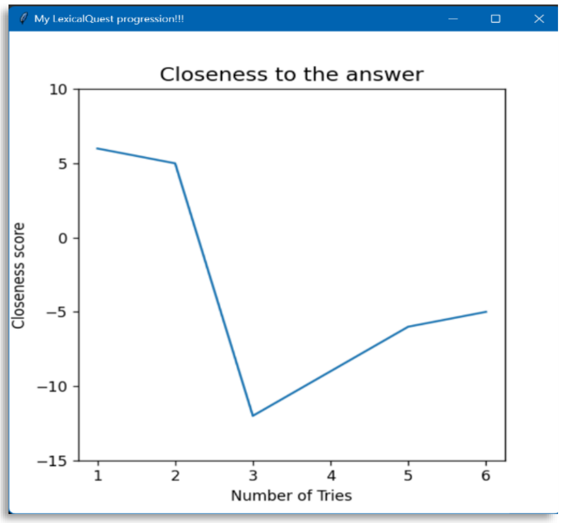
The word selection process is another crucial element of the game. The Lexi-Q utilizes a predefined set of words for each game mode. These words are loaded from text files containing ordered lists of words with the corresponding letter count. By randomly selecting a word from the appropriate word set, the game ensures that each gameplay session presents a unique word for the player to guess.



```
 H E L L O
 W O R L D
 - - - - -
 - - - - -
 - - - - -
 - - - - -

They are not in the secret word:
 ['H', 'L', 'O', 'R', 'W']

Wrong guess :( Try again...
Your score is => 11

Type your guess: create
Word must be 5 characters long.
Type your guess: hello
You've already used HELLO, please make another guess.
Type your guess: weins
```

Fg1: Terminal interface

When players enter their word guesses, the game provides feedback with different colors on the correctness of the guessed letters and their positions within the secret word. **(Fg.1)** The feedback is presented through a colored result, where correctly guessed letters in the correct positions are highlighted in green, correctly guessed letters in incorrect positions are highlighted in yellow, and incorrect letters are displayed in white. This feedback mechanism allows players to make informed decisions and adjust their subsequent guesses based on the information provided.

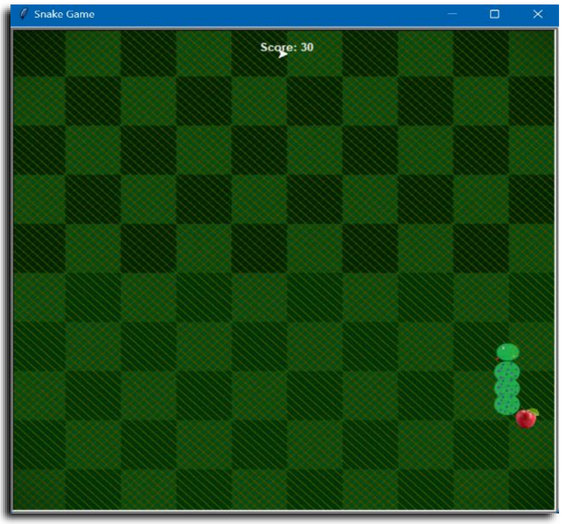


Fg2: Closeness graph

Furthermore, the Lexi-Q incorporates a scoring system that rewards players for accurate guesses while penalizing them for incorrect guesses. The scoring system takes into account the remaining attempts and assigns a score to each guess accordingly. Correct guesses with precise positions yield higher scores, while incorrect guesses or letters in incorrect positions result in deductions. This scoring mechanism adds an element of competition and motivation for players to strive for higher scores and improve their performance with each gameplay session. At the end of the gamet hay can see their progress. **(Fg2)**

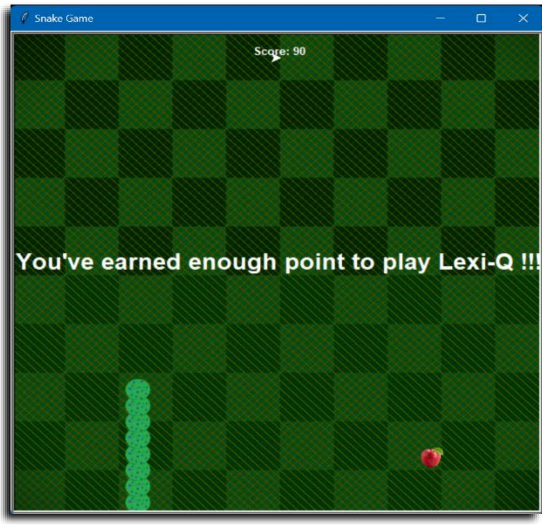
To enhance the overall gaming experience, the game also keeps track of the letters that are not present in the secret word. These letters are stored in a list and displayed to the player, providing valuable information to refine their future guesses and eliminate possibilities. This feature promotes strategic thinking and systematic deduction, encouraging players to analyze the feedback and make educated guesses based on the available information.

In addition to the main Lexi-Q, a mini-game feature has been implemented to provide an added level of entertainment and challenge. The mini-game incorporated into the Lexi-Q is similar to a classic Snake game.

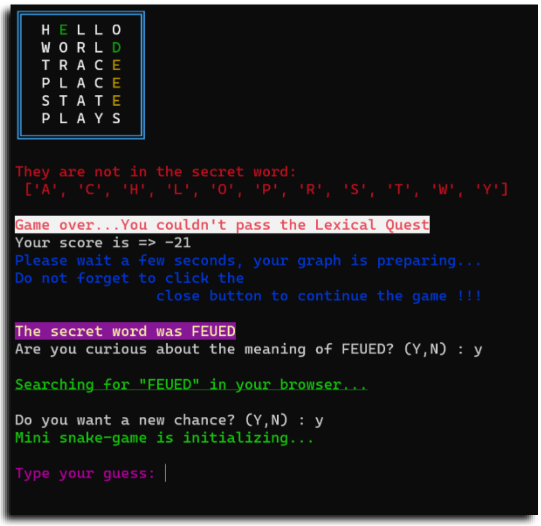


Fg3: Snake Game Turtle Interface

The Snake mini-game offers players a different gameplay experience, allowing them to control a snake that grows longer as it consumes food items. The objective of the game is to navigate the snake around the game board, avoiding collisions with walls or the snake's own body. The player earns points for each food item consumed, and the game becomes progressively more challenging as the snake grows in length. The Snake mini-game is seamlessly integrated into the Lexi-Q, allowing players to access it as an optional activity. It provides a refreshing break from word guessing and adds an element of excitement and skill-based gameplay. To implement the Snake mini-game, a separate game logic and graphical interface have been developed. The game board consists of a grid-based layout, and the snake and food items are represented by graphical elements on the screen. Players can control the snake's movement using keyboard inputs **(w,a,s,d,space)**. The mini snake game not only adds an entertaining element to the Lexi-Q Word Guessing Game but also provides an opportunity for players to take a break from word guessing. It adds depth to the overall gaming experience and ensures that players have a diverse range of activities to enjoy within the game.



You've earned enough point to play Lexi-Q !!!



```
H E L L O
W O R L D
T R A C E
P L A C E
S T A T E
P L A Y S

They are not in the secret word:
 ['A', 'C', 'H', 'L', 'O', 'P', 'R', 'S', 'T', 'W', 'Y']

Game over...You couldn't pass the Lexical Quest
Your score is => -21
Please wait a few seconds, your graph is preparing...
Do not forget to click the
                close button to continue the game !!!

The secret word was FEUED
Are you curious about the meaning of FEUED? (Y,N) : y

Searching for "FEUED" in your browser...

Do you want a new chance? (Y,N) : y
Mini snake-game is initializing...

Type your guess:
```

Fg4-5: New Game Chance

In conclusion, the Lexi-Q incorporates various elements to deliver an engaging and intellectually stimulating experience. The game mode customization, word selection process, feedback mechanism, and scoring system work together to challenge players' word-guessing abilities and promote strategic thinking. By providing informative feedback like at the end of the game if user wants s/he can see the meaning of the word and encouraging continuous improvement, the game enhances players' cognitive skills and offers an enjoyable gameplay experience.

The next section will present the conclusions drawn from the development and implementation of the Lexi-Q, summarizing the key findings and implications of the project.

Conclusions

- The Lexi-Q offers players a captivating and challenging experience, stimulating their word recognition and deductive reasoning skills. The game's customizable difficulty levels provide players with varying levels of challenge, accommodating different skill levels and preferences.

- The integration of web scraping techniques to fetch and order words from an external source enhances the game's vocabulary, ensuring a diverse and extensive word selection. The dynamic nature of the word list allows for periodic updates, keeping the game fresh and up-to-date.

- The feedback mechanism in the game, which provides colored results indicating the correctness of guessed letters and their positions, empowers players to make informed regarding their decisions and adjust subsequent guesses with supportive and positive feedback messages. This feature promotes strategic thinking and enhances the overall gameplay experience.

- The scoring system in the Lexi-Q rewards accurate guesses while penalizing incorrect ones, fostering a sense of competition and motivation for players to strive for higher scores and improve their performance.

- The inclusion of a mini-game, provides an additional layer of entertainment and challenge within the Lexi-Q. The seamless integration of the mini-game offers players a refreshing break from word guessing and adds excitement and skill-based gameplay.

In conclusion, the development and implementation of the Lexi-Q have successfully created an engaging and intellectually stimulating gaming experience. By incorporating customizable difficulty levels, a dynamic word selection process, informative feedback, and a scoring system, the game challenges players' word-guessing abilities and promotes strategic thinking. Additionally, the integration of a mini-game enhances the overall gaming experience and offers players a diverse range of activities to enjoy. The Lexi-Q effectively combines entertainment and cognitive development, providing a rewarding gameplay experience for players.

Recommendations

Word Selection: Expanding the word selection process by incorporating additional external sources, implementing a user-generated word database or dividing wordlists into categories could be a good idea for enhancing game experience.

Multiplayer Mode: Adding a multiplayer mode that allows players to compete against each other in real-time will increase engagement and provide a social aspect to the game. This can be achieved through online connectivity or local network play.

Expanding Mini-Game Options: The possibility of including more mini-games or activities within the Lexi-Q will provide even more variety and entertainment for players, allowing them to take breaks from word guessing while staying within the game environment. Dinosour game, flappy bird or other minigames could be a good.

Community Features: Incorporating community features such as leaderboards, player profiles, and social sharing options will encourage competition, foster a sense of community among players, and allow them to share their achievements with others. This will also adds a social aspect to the Lexi-Q.

Multilingual user interface and wordlists: Other languages can be added to Lexi-Q to reach global audience. This will make the Lexi-Q accessible to players from different countries and cultures, enhancing its appeal and potential user base.

References

Wordfind **=> For word database**

Merriam-Webster **=> To search the meaning of the word online**

For any further information about the game logic please take a look at the source codes