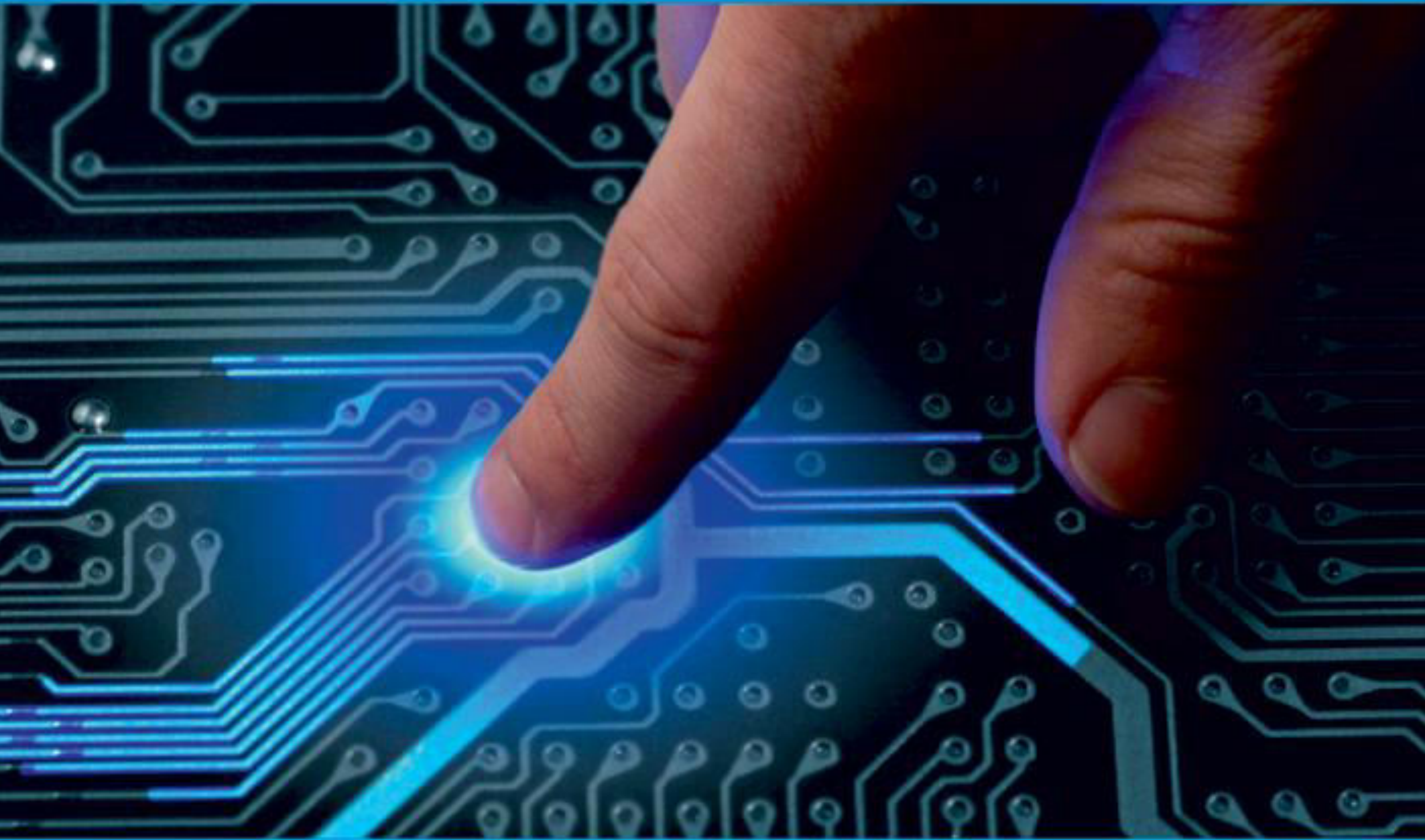
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

☎ 9940 572 462 ☎ 6381 907 438 ✉ ijircce@gmail.com @ www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204126 |

Survey on Hangman Game

Saurabh Jaiswal, Sampada Mhaske, Varad Yeola, Shraddha Jadhav, Shreyas Kolte

Prof.Vishaka Chilpipre , Prof.Trupti Khose

Student, Department of Information Technology, Dhole Patil College of Engineering Pune, Maharashtra, India

Professor, Department of Information Technology, Dhole Patil College of Engineering Pune, Maharashtra, India

ABSTRACT: The Hangman Game Project presents an engaging and interactive digital rendition of the classic word-guessing game. This endeavor seeks to create an immersive gaming experience that combines entertainment with educational elements. The project aims to develop a user-friendly interface, captivating visual design, and dynamic gameplay mechanics to serve to diverse audiences, including casual gamers and enthusiasts of word puzzles. Key features of the Hangman Game Project include a comprehensive word database encompassing a wide range of categories and difficulty levels, ensuring a varied and stimulating gameplay experience. Additionally, the project incorporates adaptive difficulty settings to tailor the challenge level to the player's proficiency, enhancing both accessibility and engagement.

KEYWORDS:
Word guessing
Puzzle game
Educational game
Language learning
Cognitive skills
Guessing game

I. INTRODUCTION

Welcome to the captivating world of Hangman! Hangman is a classic word-guessing game that has entertained generations with its simplicity and suspense. In this digital adaptation, we bring the timeless fun of Hangman to life with modern flair and engaging gameplay. The objective is straightforward: guess the hidden word before the hangman meets his fate. With each incorrect guess, a piece of the hangman is drawn, bringing players one step closer to their perilous demise. But with wit, strategy, and a little luck, victory is within reach .Our Hangman game project aims to deliver an immersive and entertaining experience for players of all ages. Whether you're a word enthusiast looking to test your vocabulary or a casual gamer seeking a quick and enjoyable challenge, Hangman promises hours of entertainment and excitement. Let's see if you have what it takes to outsmart the hangman and emerge victorious in this thrilling game of wit and strategy. Get ready to unravel the mystery, one letter at a time

II. RELATED WORKS

GAME DESIGN:

Determine the rules of the game: how many players, how to win or lose, etc.
Decide on the user interface: how the game will be displayed to the player.
Design the algorithm for selecting words to guess and revealing correct guesses.

Coding:

Set up the programming environment (e.g., IDE, programming language).
Write the code for the game logic, including:
Selecting a word for the player to guess,including;

Handling user input: guessing letters or the whole word.
Checking if the guessed letters/words are correct and updating the display accordingly.
Keeping track of the number of guesses remaining and any penalties for incorrect guesses.
Implementing a way to end the game when the word is guessed correctly or the player runs out of guesses.

Testing:

Test the game thoroughly to ensure that it works as expected.
Check for bugs, such as incorrect word selection, display errors, or game logic flaws.
Gather feedback from testers and make any necessary adjustments.

Deployment:

Package the game for distribution, depending on the platform (e.g., desktop application, web-based).
Release the game to the intended audience, either through direct download, app stores, or online Platforms.

III. PROPOSED ALGORITHM

```
import random
wList=['Earth','Fire','Wind','Water','Heart','By','your','power','combined','I','am','Captain','Planet']
ind=random.randint(0,10) # generating random integer.In exam the range was 0 to 100.But we will use 0 to 10 here
the_word_beta1=wList[ind]
the_word_beta2=the_word_beta1.lower() # making all the chars lower case for avoiding upper case-lower case match conflict
the_word=the_word_beta2.strip() # taking out all the white spaces before and after the word as there is one word per line in the words.txt. If we don't do it the length of the word will not be correct
print(the_word)
wLen=len(the_word)
u='*'*wLen
counter=0
# 12 attempts
while (counter!=12 and wLen!=0):
   counter=counter+1
   print (u)
   guess_list=list(u)
   g= input("Guess the letter: ")
   for index,letter in enumerate(the_word):
      if letter==g:
         guess_list[index] = letter
         wLen=wLen-1
   u=''.join(guess_list)

if wLen==0:
   print ("Congratulation!!!you have guessed the word")
   print ("The word was",the_word)

else:
   print ("sorry you loose")
```

IV. PSEUDO CODE

```
Step1:Generate a word at a random and store it in a variable
Step 2:  display the length of the word to the user
Step 3:  correct guesses is less than the length of the word
Step 4:  prompt the user to guess a letter
             if the guess is correct increment correct guesses by 1
             if the guess is incorrect increment incorrect guesses by 1
             and draw the next part of the hangman
Step 5: if the incorrect guesses is greater than 8, tell the user
Step 6:  they lost and exit the program).
Step 7: if correct_ guesses is equal to the length of the word, tell the user they won.
Step 8: End.
```

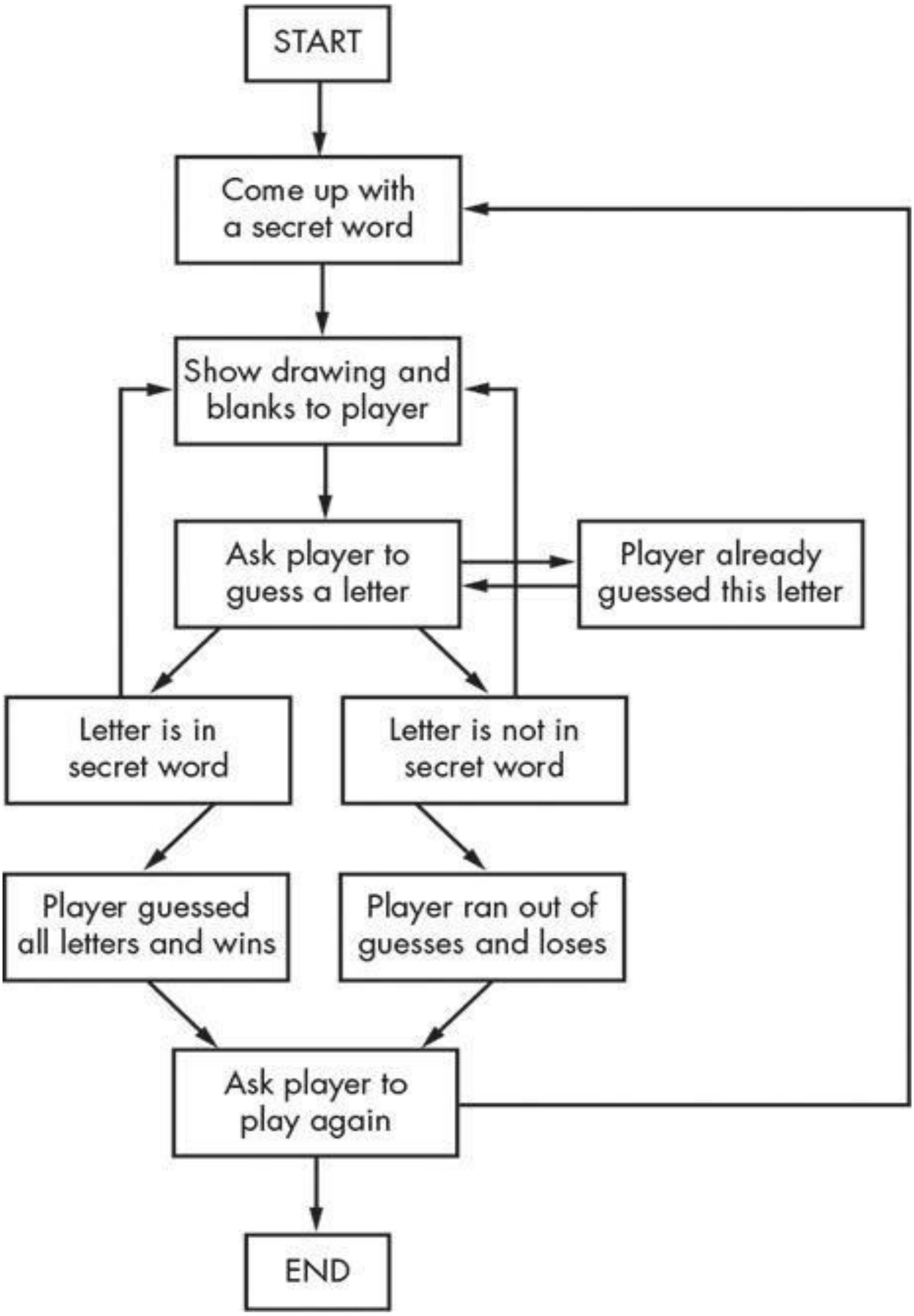
V. SIMULATION RESULTS

Hangman is a simple word guessing game where one player thinks of a word and the other player tries to guess it by suggesting letters within a certain number of guesses. Here's how you might simulate the outcome of such a project.First, you need to decide how you're going to select words for the game. You could use a predefined list of words, or you could generate random words from a dictionary or a specific category (like animals, countries, etc.) Set up the logic of the game. This includes defining the rules, such as the maximum number of incorrect guesses allowed, keeping track of guessed letters, and updating the display to show the correctly guessed letters and the progress of the word. Run simulations of the game to gather data on various aspects You can simulate different scenarios such as: Analyze the simulation results to draw insights and improve the game. For example, if certain words are consistently guessed incorrectly, they might be too difficult and should be replaced with easier ones. If players frequently guess certain letters, you might want to adjust the word selection process to include words that use a wider range of letters. Based on the analysis, make adjustments to the game logic or word selection process and run the simulation again to see how the changes affect the outcome Once you're satisfied with the simulation results, validate them by testing the game with real players to see if the simulated outcomes match real-world results Simulate how players interact with the game interface, including entering guesses and receiving feedback.Simulate scenarios where players enter invalid inputs (e.g., non-alphabetic characters) and observe how the game responds. Experiment with different difficulty levels by adjusting parameters such as word length or the number of allowed incorrect guesses per game

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204126 |


START
Come up with a secret word
Show drawing and blanks to player
Ask player to guess a letter
Player already guessed this letter
Letter is in secret word
Letter is not in secret word
Player guessed all letters and wins
Player ran out of guesses and loses
Ask player to play again
END

.

VI. CONCLUSION AND FUTURE WORK

Conclusion: The Hangman game project has been successfully implemented, providing an entertaining and interactive experience for users. Through this project, we've demonstrated proficiency in programming concepts such as data structures, user input handling, random word selection, and game logic implementation. The game offers a classic yet engaging way to improve vocabulary and problem-solving skills

Future Work: To further enhance the Hangman game, several avenues of improvement can be explored. Firstly, incorporating a graphical user interface (GUI) can enhance the user experience by providing visual feedback and a more immersive interface. Additionally, expanding the word database and implementing difficulty levels can cater to a wider audience and increase replay value. Integrating multiplayer functionality would enable users to compete with friends or other players online, adding a social aspect to the game. Furthermore, incorporating features like hints, power-ups, or themes can add depth and variety to the gameplay experience. Lastly, optimizing the codebase for performance and scalability can ensure smooth operation across different platforms and devices. Overall, there are numerous opportunities for further development and refinement of the Hangman game to make it even more enjoyable and engaging for players.

.

REFRENCES

1. United Nations, “Transforming our world: The 2030 agenda for sustainable development,” United Nations, General Assem. Resolution, New York, NY, USA, Tech. Rep. A/RES/70/1, 2015.
2. K. Almohammadi, H. Hagras, D. Alghazzawi, and G. Aldabbagh, “A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms,” J. Artif. Intell. Soft Comput. Res., vol. 7, no. 1, pp. 47–64, Jan. 2017, doi: 10.1515/jaiscr-2017-0004.
3. J. S. Wu and J. J. Lee, “Climate change games as tools for education and engagement,” Nature Climate Change, vol. 5, pp. 413–418, Apr. 2015, doi: 10.1038/NCLIMATE2566.
4. M. R. Ryan and E. L. Deci, “Intrinsic and extrinsic motivations: Classic definitions and new directions,” Contemporary Educ. Psychol., vol. 25, no. 1, pp. 54–67, Jan. 2000, doi: 10.1006/ceps.1999.1020.
5. T. W. Malone and M. R. Lepper, “Making learning fun: A taxonomy of intrinsic motivations for learning,” Aptitude, Learn. Instruction, vol. 3, pp. 223–253, 1987. .
6. A. Handy, “Interview: Markus ‘notch’ persson talks making Minecraft,” Mar. 2010. Accessed: Oct. 30, 2017[Online.]Available:https://www.gamasutra.com/view/news/27719/Interview_Markus_Notch_Persson Talks_Making_Minecraft
7. [D. Heaven, “When infinity gets boring: What went wrong with No Man’s Sky,” Sep. 2016. Accessed: Oct. 30, 2017. [Online.] Available: https://www.newscientist.com/article/2104873-when-infinity-getsboring-what-went-wrong-with-no-mans-sky/
8. M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, “Procedural content generation for games: A survey,” ACM Trans. Multimedia Comput., Commun. Appl., vol. 9, no. 1, 2013, Art. no. 1
9. . S. Colton, “Automated puzzle generation,” in Proc. AISB02 Symp. AI Creativity Arts Sci., 2002, pp. 99–108
10. N. Shaker, J. Togelius, and M. J. Nelson, Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Berlin, Germany: Springer, 2016.
11. S. Glickman, N. McKenzie, J. Seering, R. Moeller, and J. Hammer, ‘‘Design challenges for livestreamed audience participation games,’’ in Proc. Annu. Symp. Comput.-Hum. Interact. Play Extended Abstr. (CHI PLAY), Oct. 2018, pp. 187–199.
12. Y. Jiang, P. Paliyawan, R. Thawonmas, and T. Harada, ‘‘An audience participation angry birds platform for social well-being,’’ in Proc. 19th Annu. Eur. Game-On Conf. Simulation AI Comput. Games (GAMEON), 2018, pp. 1–6

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

INNO SPACE
SJIF Scientific Journal Impact Factor

doi
crossref

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details