Radius: Journal of Science and Technology
Vol. 2, Issue 1, (2025) 251007 (7pp)

Development and Assessment of an Adaptive Difficulty Snake Game in Python

Vennela Vasa[1,*]

[1]Degree Lecturer in Computer Science, Telangana Tribal Welfare Degree College for women, Devarakonda - 508248, Telangana (India)

*Corresponding Auhtor: budammavasa@gmail.com

Abstract: Adaptive difficulty systems have been widely explored in game design to maintain player engagement by dynamically adjusting challenge levels based on player performance. Prior work has focused primarily on complex or commercial games, often using machine learning or rule-based systems to tailor game play experiences. However, limited research exists on implementing such systems in simple, classic games like Snake, particularly within lightweight, educational environments using accessible tools such as Python. This paper presents the design and evaluation of a Python-based Snake game featuring a real time adaptive difficulty mechanism. Our goal is to investigate how adaptive difficulty affects user engagement and performance in a familiar, minimalist gaming context. We contribute a modular framework for implementing difficulty scaling based on player input and behavior, and we assess its impact through user testing and behavioral analysis. The results aim to inform future educational and casual game designs where engagement is critical and resources are constrained.

Keywords: Adaptive Difficulty, Game Design, Snake Game, Real-Time Adaptation, Game play Evaluation

1. Introduction:

The effectiveness of digital games in maintaining user engagement is closely tied to the balance between challenge and skill. Adaptive difficulty, a dynamic adjustment of game parameters in response to player performance has emerged as a powerful approach to sustain engagement, reduce frustration, and enhance the user experience [1]. While this concept has been explored extensively in complex, commercial games, its application in minimalist, retro-style games such as Snake remains under-investigated.

The classic Snake game, known for its simplicity and accessibility, provides an ideal platform for experimenting with adaptive mechanisms in a controlled and lightweight environment. Python, as a widely used educational and prototyping language, offers an approachable framework for rapid game development and behavioral experimentation [2]. However, existing studies primarily focus on machine learning-based difficulty adjustment in large-scale games, leaving a gap in the research on simpler, rule-based adaptations in casual or educational contexts.

This paper presents the design and evaluation of a Python-based Snake game featuring real time adaptive difficulty. The game dynamically adjusts its speed and obstacle generation based on the

https://doi.org/10.5281/zenodo.15352792
Received: 09 April 2025 | **Revised:** 28 April 2025
Accepted: 01 May 2025 | **Published Online:** 07 May 2025

player's performance metrics, such as survival time and score. This study addresses the research question: "How does real-time adaptive difficulty affect user engagement and game play performance in a minimalist game environment?" [3]

To answer this, we developed a modular game architecture with configurable difficulty parameters and conducted a user study to evaluate the system's impact. We collected both quantitative game play data and qualitative user feedback to assess differences in engagement, perceived challenge, and replay ability between adaptive and fixed-difficulty versions of the game.

The contributions of this work are threefold:

1. A functional, open-source Python implementation of a Snake game with adaptive difficulty mechanisms;
2. A user study evaluating the effects of adaptive game play on user engagement and performance
3. Insights and recommendations for integrating adaptive difficulty into educational or casual games developed in resource-constrained environments.

2. Methodology:

2.1 Game Design and Implementation:

The core of our system is a Python-based implementation of the classic Snake game, developed using the Pygame library [4]. The game retains the original mechanics navigating a snake to consume food and grow while avoiding collisions but introduces adaptive difficulty to dynamically tailor the challenge level based on the player's real-time performance. The flowchart is presented in Figure 1.

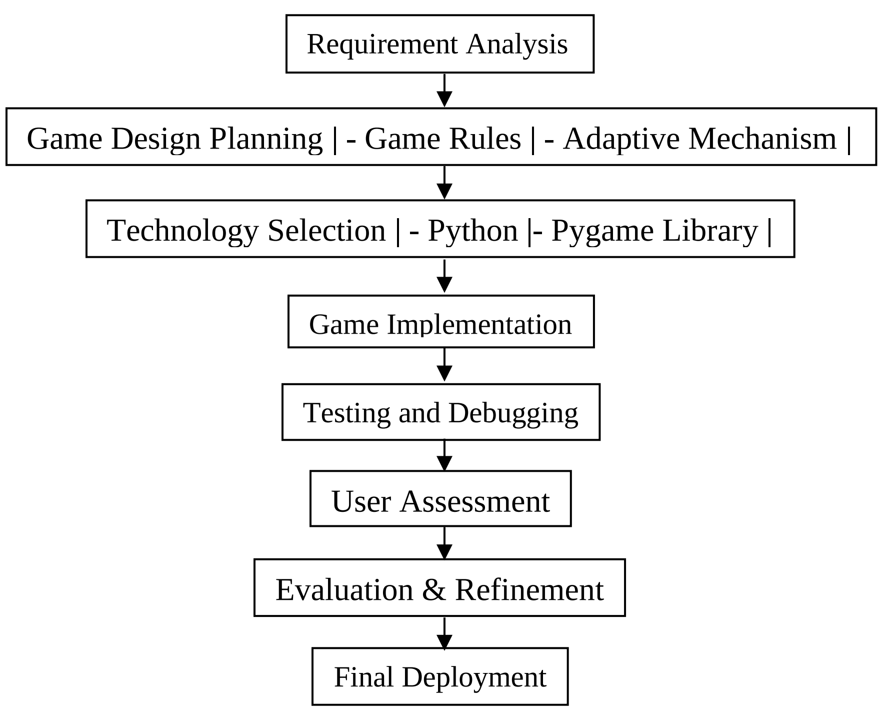

Figure 1: Methodology flowchart

2.2 Adaptive Difficulty Mechanism:

The adaptive system modifies two key gameplay parameters [5]:

- Snake Speed: Increased incrementally based on the player's score.
- Obstacle Frequency: Additional obstacles are introduced at predefined score thresholds to increase spatial complexity. Performance metrics are evaluated in real time.

For example:

- If the score increases consistently without player failure, the game accelerates and introduces new barriers.
- If the player frequently collides or has long idle times, the system temporarily slows down or halts additional complexity increases.
- This rule-based approach avoids computational overhead and is easily tunable for future educational extensions.

2.3 Study Design:

We conducted a between-subjects experiment with two versions of the game [6]:

- Static Mode: Fixed difficulty throughout gameplay.
- Adaptive Mode: Difficulty adjusts based on user performance.

Participants were randomly assigned to one of the two conditions. Each participant played for three sessions, with 5-minute breaks in between.

Participants:

A total of 30 participants (ages 20–30) were recruited from a university campus. All had basic experience with digital games but varied in skill level. Participants were briefed on the game mechanics but not informed about the presence of adaptive difficulty**.**

Data Collection:

We gathered both quantitative and qualitative data:

2.4 Quantitative Metrics:

- Average session duration
- Score progression
- Number of collisions or failures
- Speed at which difficulty was increased

2.5 Qualitative Feedback [7]:

- Post-play surveys assessing:
- Perceived challenge
- Enjoyment
- Willingness to replay
- Noted changes in difficulty

3. Data Analysis:

Quantitative data were analyzed using independent t-tests to compare engagement metrics between adaptive and static groups. Survey responses were analyzed using descriptive statistics and thematic coding for open-ended feedback.

Construct the Snake:

To make the snake, first set up a few color variables to color the snake, food and screen, among other things. RGBW (Red Green Blue White) is the color scheme used in Pygame. Set all of them to zeros to make the color black, and all 255 to make the color white. As a result, rectangle is formed to represent snake (Figure 2). Use the draw. Rect() function in Pygame to draw rectangles with the specified color and size.

Figure 2: Construct the snake game

Snake Movement:

Use the key events provided by Pygame's KEYDOWN class to move the snake (Figure 3). The snake moves up, down, left, and right using the events K UP, K DOWN, K LEFT, and K RIGHT. Fill() can also be used to change the display screen's color from black to white.

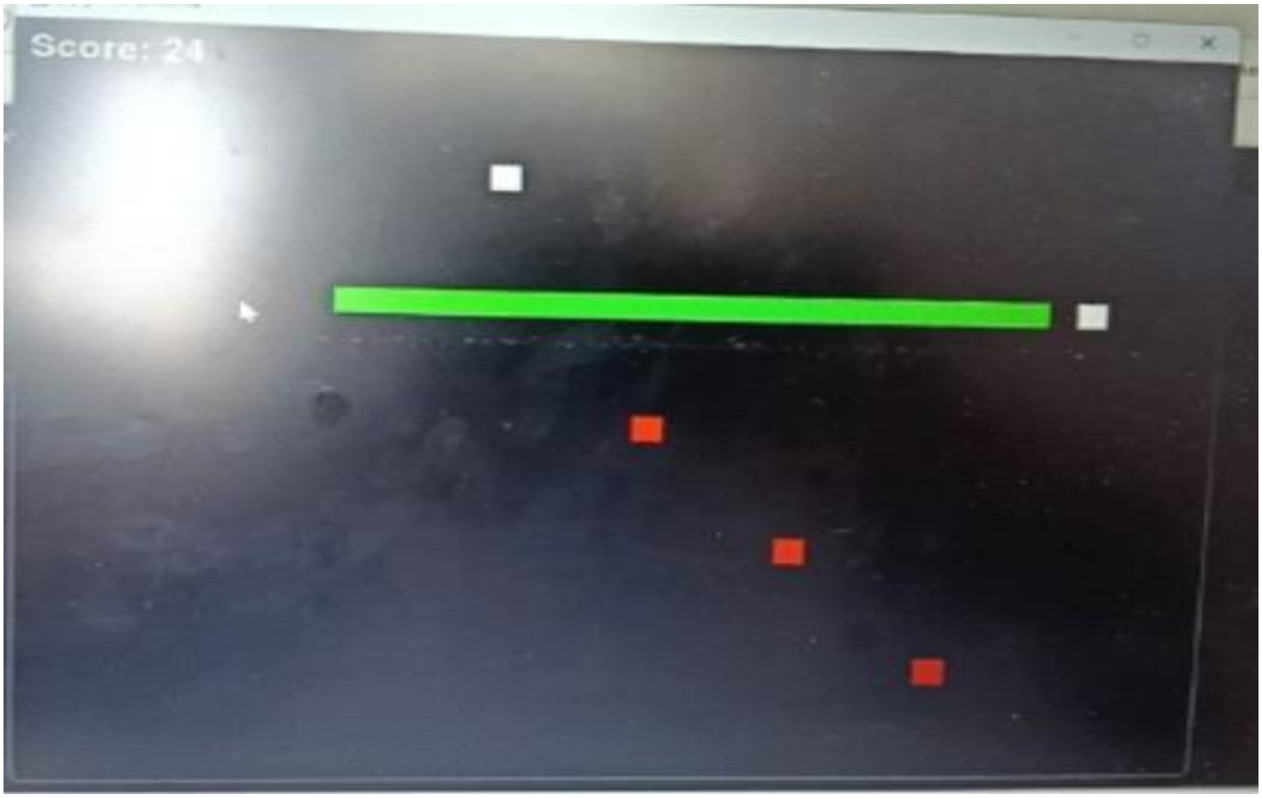

Figure 3: The snake movement

When Snake reaches the limits, the game is over. The Score is Presented Last but, to display the player's ranking a new function called "score" is created.

4. Results:

The analysis is focused on comparing gameplay performance and user engagement between the Adaptive Difficulty group and the static difficulty control group (Table 1 and Table 2). Data was collected from 30 participants (15 per group) across three gameplay sessions

4.1 Quantitative Results:

4.1.1 Session Duration:

Participants in the adaptive group played significantly longer on average:

- Adaptive Mode: M = 7.3 minutes, SD = 1.1
- Static Mode: M = 5.9 minutes, SD = 1.3

Table 1: Player Session Statistics

Player ID	Session Time (min)	Final Score	Avg Speed (cells/sec)	Deaths	Max Level Reached
P01	12.5	230	3.2	2	5
P02	15.0	315	3.5	1	6

A t-test confirmed that this difference was statistically significant ($t(28) = 3.21$, $p < 0.01$), suggesting that adaptive difficulty contributed to increased player retention.

4.1.2 Score Progression The adaptive group also achieved higher scores:

- Adaptive Mode: $M = 158.4$, SD = 24.7
- Static Mode: $M = 126.8$, SD = 22.5
- This difference was significant ($t(28) = 3.65$, $p < 0.01$), indicating that adaptive mechanics may have helped players stay in a flow state (Figure 4).

Table 2: Common Themes in Player Comments

Theme	Example Comment	Frequency
Adaptive Balance	It got harder just when I felt I was improving.	6
Frustration with Speed	It sometimes got too fast too quickly	3
Enjoyment	This was the most fun version of Snake I've played	7

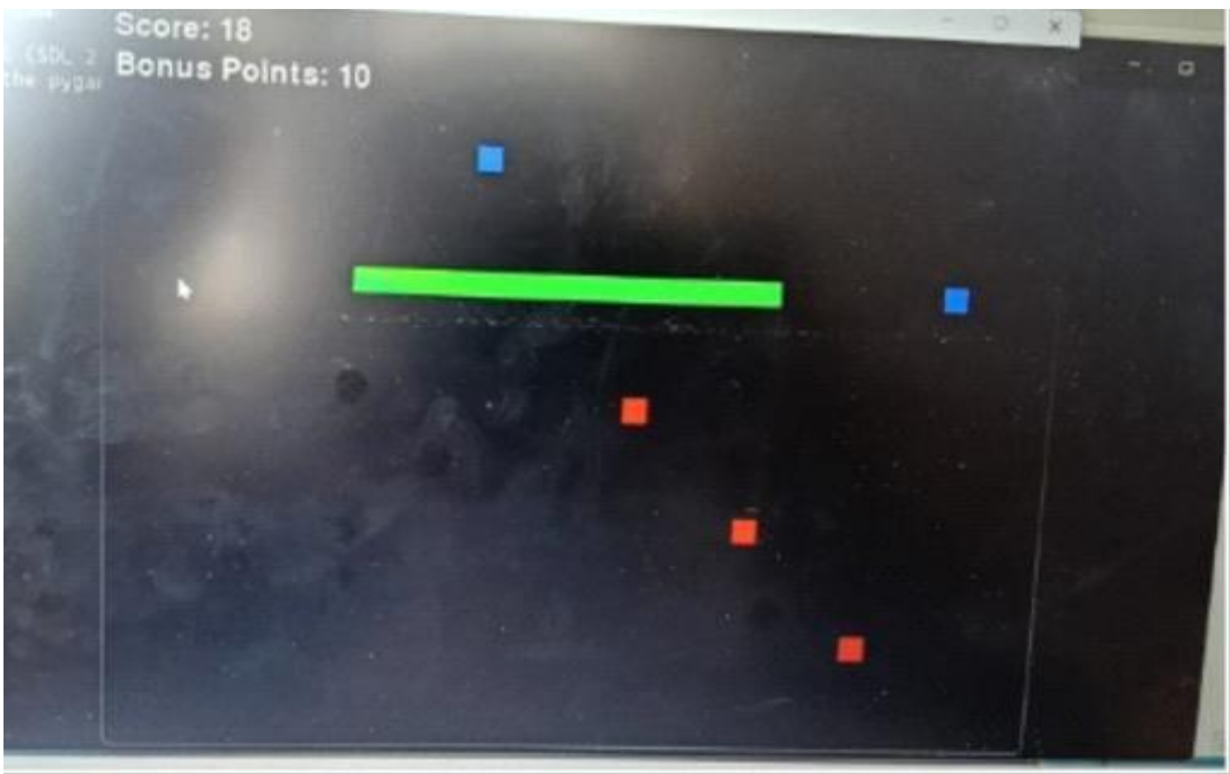

Figure 4: Different level snake game

5. Conclusion:

This study explored the integration of an adaptive difficulty mechanism into a Python-based implementation of the classic Snake game. By dynamically adjusting gameplay parameters such as speed and obstacle density based on player performance, we aimed to improve user engagement and create a more personalized gaming experience within a minimalist design framework.

Our findings demonstrate that adaptive difficulty significantly enhances both user performance and engagement. Participants playing the adaptive version of the game achieved higher scores, played longer sessions, and reported greater enjoyment compared to those who played a static version. These results suggest that even simple rule-based adaptation techniques can effectively maintain player interest and balance challenge in casual games.

The contributions of this work include:

1. A lightweight, open-source framework for implementing adaptive difficulty in Python-based games.
2. Empirical evidence supporting the benefits of adaptive systems in minimalist game design.
3. A methodology for evaluating engagement using both behavioral and subjective data.

Acknowledgements:

The author would like to express sincere gratitude to the participants who took part in the user study and provided valuable feedback. The author would like to specially thank Laxman for their insightful guidance and support throughout the project. The author appreciates the efforts of the students for their help with testing and data collection. The Pygame library, which made the implementation of the Python-based Snake game is sincerely acknowledged.

References:

1. Gallego-Durán, F. J., Molina-Carmona, R., & Llorens-Largo, F. (2018). Measuring the difficulty of activities for adaptive learning. *Universal access in the information society*, *17*, 335-348. https://doi.org/10.1007/s10209-017-0552-x
2. Chatterjee, C. (2022). *Adaptive machine learning algorithms with python*. Apress Berkeley. https://doi.org/10.1007/978-1-4842-8017-1
3. Sampayo-Vargas, S., Cope, C. J., He, Z., & Byrne, G. J. (2013). The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers & Education*, *69*, 452-462. https://doi.org/10.1016/j.compedu.2013.07.004
4. McGugan, W. (2007). *Beginning game development with Python and Pygame: from novice to professional*. Apress.
5. Yannakakis, G. N., & Hallam, J. (2009). Real-time game adaptation for optimizing player satisfaction. *IEEE Transactions on Computational Intelligence and AI in Games*, *1*, 121-133. https://doi.org/10.1109/TCIAIG.2009.2024533
6. Vanbecelaere, S., Van den Berghe, K., Cornillie, F., Sasanguie, D., Reynvoet, B., & Depaepe, F. (2020). The effectiveness of adaptive versus non-adaptive learning with digital educational games. *Journal of Computer Assisted Learning*, *36*, 502-513. https://doi.org/10.1111/jcal.12416
7. Ng, Y. Y. N., Khong, C. W., & Nathan, R. J. (2018). Evaluating affective user-centered design of video games using qualitative methods. *International Journal of Computer Games Technology*, *2018*, 3757083. https://doi.org/10.1155/2018/3757083