ANIMATION GAME ENGINE FOR INTERACTIVE PRESENTATION OF EDUCATIONAL MEDIA

Ashwin Ravi*, Matli Mithilesh Reddy†, Dr. Sakaya Milton R‡

IV Year, Department of Computer Science and Engineering
SSN College of Engineering, India
Email: *ashwin2110720@ssn.edu.in, †mithileshreddy2111010@ssn.edu.in

‡Professor, Department of Computer Science and Engineering
SSN College of Engineering, India
Email: miltonrs@ssn.edu.in

***Abstract*—This paper presents an idea for interactive animation engine developed using Godot Engine and GDScript to enhance the visualization and understanding of complex educational concepts. Traditional teaching methods often struggle to effectively convey dynamic processes in subjects like mathematics, science, and computing, as static materials lack interactivity and fail to illustrate step-by-step transformations. To address these limitations, the proposed engine integrates keyframe-based animations, interactive elements, and real-time feedback, enabling educators to create engaging and adaptive learning experiences without requiring advanced programming expertise. By supporting personalized learning paths, user interaction tracking, and accessibility across various devices, this tool empowers educators to present abstract concepts more intuitively. Its versatile applications span multiple disciplines, including algorithm visualization, physics simulations, and biological processes, making it a powerful tool for bridging the gap between theoretical learning and practical understanding. Through this approach, the animation engine transforms traditional learning methodologies into more engaging, interactive, and effective educational experiences.**

I. INTRODUCTION

In modern education, effectively communicating complex concepts in subjects like mathematics, science, and computing demands more than static materials; interactive and dynamic visualizations play a crucial role in enhancing comprehension and retention. Traditional educational approaches, while valuable, often rely heavily on text-based explanations and static images, which can make it challenging for learners to grasp abstract ideas or visualize step-by-step processes.

This project introduces an Animation Game Engine developed using the Godot Engine and GDScript, specifically designed to revolutionize the way educational content is delivered. The engine empowers educators to craft engaging, interactive presentations without requiring advanced programming skills. It integrates features such as keyframe-based animations, real-time quizzes, and feedback systems, making it easier to explain intricate topics through dynamic visuals and interactive activities.

One of the standout aspects of this engine is its focus on learner engagement. By enabling user-driven interactions and tracking learner progress, the system personalizes the learning journey, providing valuable insights into each student's comprehension level and areas requiring reinforcement. This not only aids educators in adapting their teaching strategies but also motivates learners to actively participate in their learning process.

Moreover, the engine's lightweight and resource-efficient design ensures accessibility across a wide range of devices, making it suitable for educational institutions with limited computing resources. Its scalability allows for broad applications, from primary education to specialized training in higher education and professional settings. In essence, this animation engine represents a significant advancement in educational technology, offering a versatile and powerful tool for delivering intuitive, engaging, and effective teaching content in a digital format.

II. LITERATURE SURVEY

1) **The Efficacy of Animation and Visualization in Teaching Data Structures: A Case Study**
   Authors: Genady Kogan, Hadas Chassidim, Irina Rabaev (2024)
   This paper highlights the effectiveness of animation and visualization techniques in improving student engagement and retention when teaching data structures. It demonstrates how visual aids help bridge the gap between theoretical concepts and practical understanding.
2) **Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings**
   Authors: Ladislav Végh, Veronika Stoffová (2017)
   This research focuses on using algorithm animations to teach basic sorting algorithms. The study empha-

sizes how visualization helps make abstract ideas more accessible and improves learners' comprehension and problem-solving skills.

3) **GILP: An Interactive Tool for Visualizing the Simplex Algorithm**
   Authors: Henry W. Robbins, Samuel C. Gutekunst, David B. Shmoys, David P. Williamson (2023)
   This paper introduces GILP, an interactive tool for visualizing the Simplex algorithm. It demonstrates how graphical representations enhance understanding by allowing learners to engage with algorithmic processes dynamically.
4) **A Review of The Algorithm Visualization Field**
   Authors: Sarthak Goel, Vanshika Varshney, Shubham Dikshant, Akhil Sharma, Sherish Johri (2023)
   This review paper surveys various algorithm visualization tools and methodologies, reinforcing the effectiveness of well-structured visual aids in improving learners' problem-solving abilities and conceptual understanding.
5) **Augmented Reality with Algorithm Animation and Their Effect on Students' Emotions**
   Authors: Maximiliano Paredes-Velasco, J. Ángel Velázquez-Iturbide, Mónica Gómez-Ríos (2022)
   This study explores the use of augmented reality (AR) combined with algorithm animation to enhance student engagement. It finds that AR-based visualizations positively influence cognitive load management and emotional response.
6) **Deep Reinforcement Learning with Godot Game Engine**
   Authors: Mahesh Ranaweera, Qusay H. Mahmoud (2024)
   This research examines integrating deep reinforcement learning within the Godot Engine to create intelligent, adaptive learning experiences, emphasizing the engine's flexibility in educational applications.
7) **Using Student-Built Algorithm Animations as Learning Aids**
   Author: John T. Stasko (1997)
   This foundational study encourages learners to create their own algorithm visualizations, leading to deeper engagement and improved conceptual grasp of computational problems.
8) **Visualizations and Animations in Learning Systems**
   Author: Andreas Kerren (2012)
   This paper advocates for integrating animations and visualizations across various educational domains, emphasizing their adaptability and benefits in learning environments.

III. PROPOSED SYSTEM

The proposed system is an advanced animation game engine specifically designed to elevate the delivery of educational content. The primary focus of this system is to transform conventional, static learning experiences into highly interactive, engaging, and adaptive environments that foster deeper understanding and active learner participation. Leveraging the Godot Engine and its flexible scripting language GDScript, the system provides educators with a powerful, user-friendly platform to create visually rich and pedagogically effective content.

A key challenge in the current educational technology landscape is the high technical expertise required to develop interactive and animated learning materials. This engine addresses that gap by offering intuitive tools that simplify the process of crafting dynamic keyframe-based animations, integrating real-time interactivity, and tracking user engagement. Through its modular design and accessible scripting interface, the engine lowers the technical barrier for educators, allowing them to focus on the pedagogical aspects of content delivery rather than the complexities of software development.

Furthermore, the system incorporates mechanisms to capture and analyze learner interactions. Data such as quiz responses, engagement patterns, and real-time input are recorded and processed to provide valuable feedback to both educators and learners. This data-driven approach supports personalized learning by identifying individual learner progress, strengths, and areas needing improvement. The engine's lightweight and optimized design ensures compatibility with a wide range of hardware platforms, making it accessible to institutions with varying technological resources.

To further enhance usability, the system is designed with a well-structured interface that streamlines the creation of interactive content while maintaining efficiency. The implementation of keyframe-based animation techniques allows educators to develop engaging visual elements without extensive coding expertise. Additionally, the real-time feedback mechanism ensures that students receive immediate responses to their interactions, reinforcing learning outcomes. The integration of various input methods, such as keyboard, mouse, and touch-based interactions, provides a versatile experience that accommodates different learning environments. By focusing on accessibility, ease of use, and interactive engagement, this system aims to bridge the gap between technology and education, fostering a more immersive and effective learning experience.

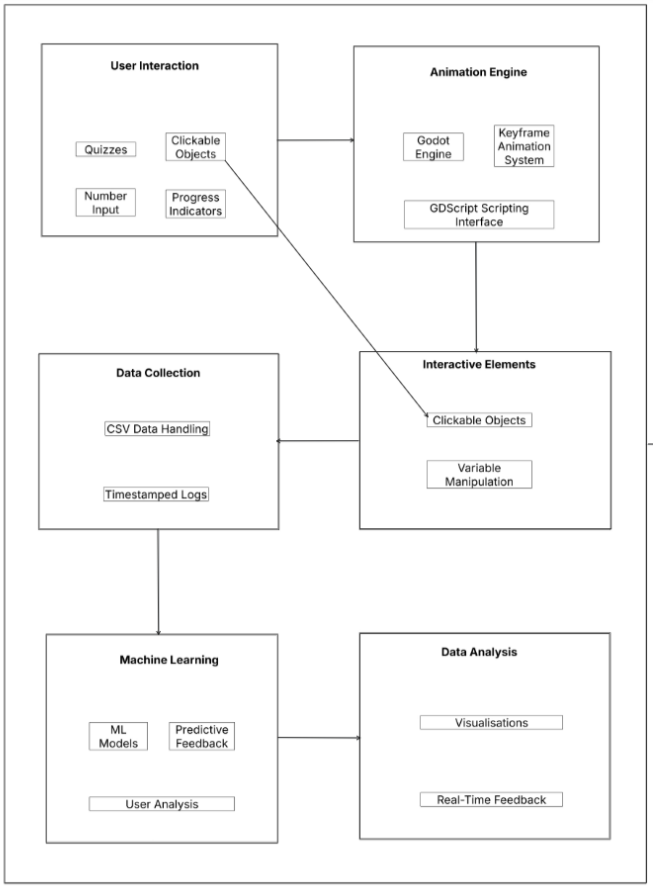

Fig. 1. Architecture Diagram

IV. IMPLEMENTATION

The implementation of the animation game engine is carried out using the Godot Engine, which offers a modular, scene-based architecture. The scripting is done using GDScript, allowing easy control over animations, interactive features, and data collection. Each module is implemented with clarity, ensuring educators can customize or extend the engine's capabilities without deep technical expertise.

A. Initialization and Setup

The initial setup of the engine involves configuring the project settings, preparing the scene hierarchy, and defining global resources. The following steps outline the setup process:

- Creating a root `Main Scene` containing primary UI elements, animation objects, and input handlers.
- Setting up `Control Nodes` for organizing user interface components like buttons, quiz panels, and progress indicators.
- Defining singleton `autoload` scripts for managing global variables, user data, and settings.
- Loading essential assets such as textures, fonts, and animations during the `ready()` function of the root scene.

Listing 1. Sample GDScript For Initialization

```
func _ready():
    load_resources()
    setup_ui()
    initialize_variables()
```

B. Keyframe Animation System

Godot's `AnimationPlayer` node is utilized to create smooth, movie-quality animations. Educators define keyframes visually through Godot's editor or programmatically via GDScript.

- Animations control properties such as position, scale, rotation, opacity, and more.
- Animation tracks can be added for multiple objects within a scene.
- The engine interpolates between keyframes to generate fluid transitions.

Listing 2. Sample GDScript For Animation Playback

```
func play_animation(anim_name):
    $AnimationPlayer.play(anim_name)
```

C. GDScript Scripting Interface

The scripting interface allows educators to embed interactivity and control animation flow without complex programming:

- Signal connections handle user interactions like button presses, mouse clicks, or touch gestures.
- Quiz logic, variable manipulation, and real-time feedback are defined via simple GDScript functions.
- State management is implemented using enums or finite state machines to control different stages of the presentation.

Listing 3. Sample GDScript For Quiz Response Handling

```
func _on_QuizButton_pressed(answer):
    if answer == correct_answer:
        show_feedback("Correct!")
    else:
        show_feedback("Incorrect!")
```

D. Data Collection

One of the key features of the engine is the real-time tracking of learner interactions. GDScript is used to record events such as quiz responses, time spent on tasks, and engagement metrics.

- Data is stored in structured CSV format using Godot's `File` class.
- Each interaction is timestamped, providing detailed logs for analysis.
- Data can be exported at the end of the session for further machine learning processing.

Listing 4. Sample GDScript For Writing Data to CSV

```
func record_data(interaction_type,
result):
    var file = File.new()
    file.open("user://data.csv",
    File.WRITE_APPEND)
    var ts = OS.get_system_time_secs()
    file.store_line(str(timestamp) +
    "," + interaction_type +
    "," + result)
    file.close()
```

E. Machine Learning Integration

Though the engine itself is implemented in Godot, the collected data is exported and used externally for machine learning analysis. Python-based models (such as Decision Trees, Random Forests) process the CSV data to evaluate learner performance and provide insights.

- After exporting, the CSV data is fed into machine learning pipelines.
- Classification and regression models are applied to predict learner progress and recommend feedback.
- Results of the analysis can be re-imported to adapt future sessions.

F. Executable Export

The Godot Engine's export functionality allows the entire project to be compiled and distributed as a standalone executable:

- The engine is exported as an `.exe` file for Windows, ensuring educators and learners can run the presentation without needing the Godot environment.
- Export templates provided by Godot ensure compatibility across multiple platforms (Windows, Linux, macOS).

V. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

A. Dutch National Flag

The DNF Problem model achieved excellent accuracy, with an R-squared score of 1.00, indicating near-perfect prediction capability. The analysis shows the Random Forest Regressor effectively identifying whether the color sorting was successful. The model's error metrics, such as Mean Absolute Error (MAE) of 0.06 and Root Mean Squared Error (RMSE) of 0.12, suggest a high level of precision in score prediction. Predicted entries for players exhibit score variations ranging from 2 to 9, accompanied by remarks like "Correct! The colors are properly sorted!" and "Not quite right. Keep trying!" Despite occasional fluctuations, the model consistently offers valuable insights into sorting accuracy and performance trends.

TABLE I
PERFORMANCE METRICS FOR DUTCH NATIONAL FLAG ALGORITHM USING RANDOM FOREST REGRESSOR

Metric	**Value**
Mean Absolute Error (MAE)	0.06
Mean Squared Error (MSE)	0.01
Root Mean Squared Error (RMSE)	0.12
R-squared (R^2 Score)	1.00

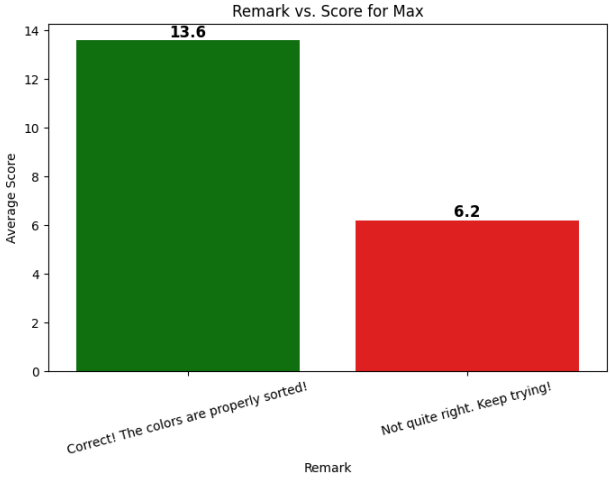

Fig. 2. Remark vs Score For Max In DNF

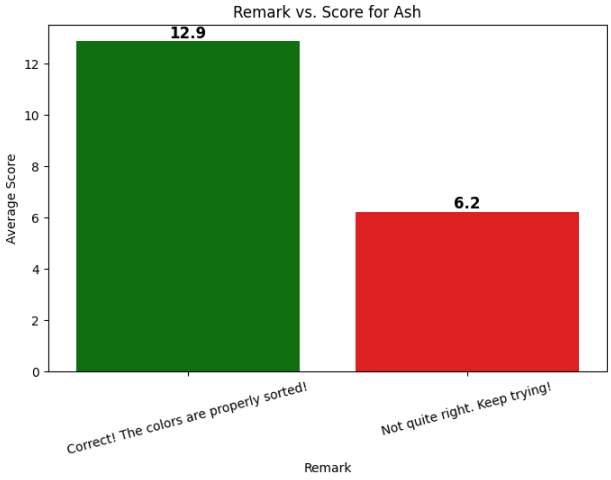

Fig. 3. Remark vs Score For Ash In DNF

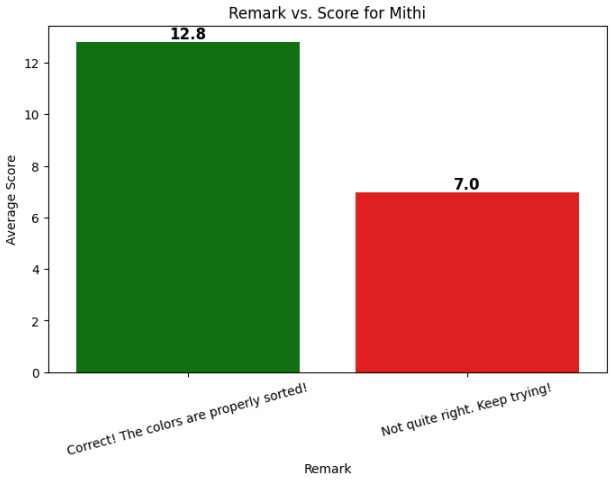

Fig. 4. Remark vs Score For Mithi In DNF

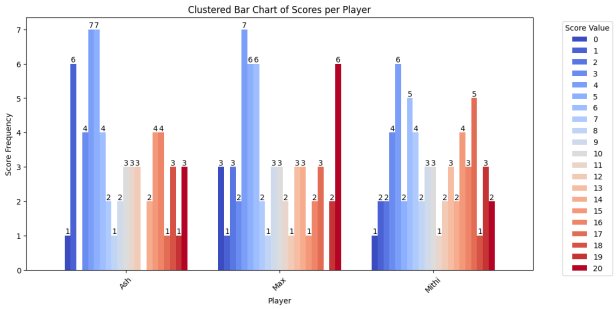

Fig. 5. Clustered Bar Chart Of Scores Per Player In DNF

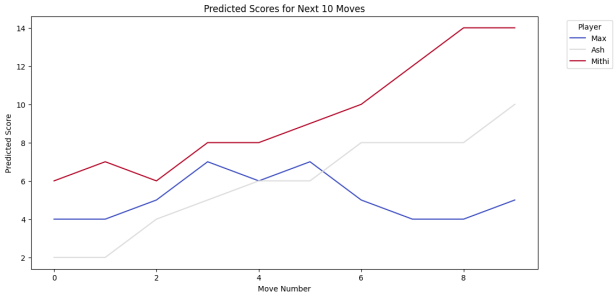

Fig. 6. Predicted Scores For Next 10 Moves (Dynamic) In DNF

B. *Towers of Hanoi*

The Tower of Hanoi model was evaluated using Decision Tree Classifier and Support Vector Machine (SVM) Classifier approaches. The Decision Tree Classifier achieved high classification accuracy of 95% , correctly distinguishing between ”Solved!” and ”Keep trying” outcomes in most cases. The classification report indicates strong precision and recall for the ”Keep trying” class, with slightly lower recall for ”Solved!” cases. In contrast, the SVM Classifier demonstrated lower accuracy of 43%, with inconsistent performance, particularly in identifying ”Solved!” cases. Overall, the Decision Tree Classifier outperformed the SVM Classifier, providing reliable classifications of learner performance across varying disk counts.

TABLE II
CLASSIFICATION REPORT FOR TOWER OF HANOI USING DECISION TREE CLASSIFIER

Decision Tree Classifier				
Class	**Precision**	**Recall**	**F1-Score**	**Support**
Keep trying	0.94	1.00	0.97	17
Solved!	1.00	0.75	0.86	4
Accuracy	-	-	0.95	21
Macro avg	0.97	0.88	0.91	21
Weighted avg	0.96	0.95	0.95	21

TABLE III
CLASSIFICATION REPORT FOR TOWER OF HANOI USING SVM CLASSIFIER

SVM Classifier				
Class	**Precision**	**Recall**	**F1-Score**	**Support**
Keep trying	0.86	0.35	0.50	17
Solved!	0.21	0.75	0.33	4
Accuracy	-	-	0.43	21
Macro avg	0.54	0.55	0.42	21
Weighted avg	0.73	0.43	0.47	21

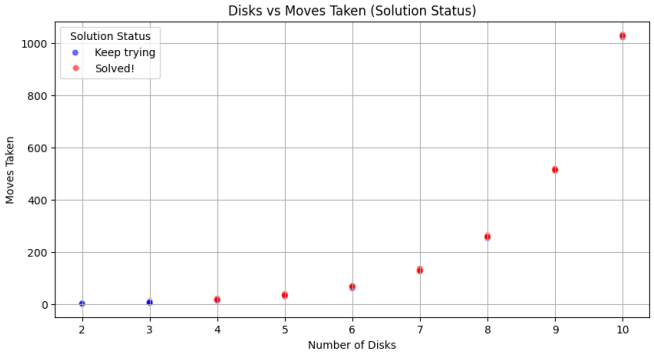

Fig. 7. Disks vs Moves Taken In Towers Of Hanoi

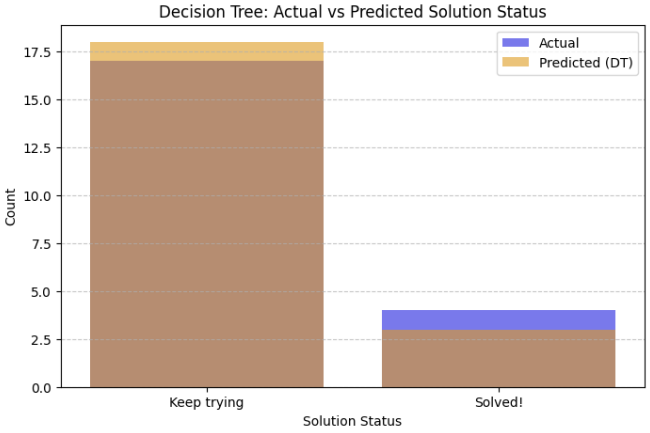

Fig. 8. Actual vs Predicted Solution Status In Towers Of Hanoi

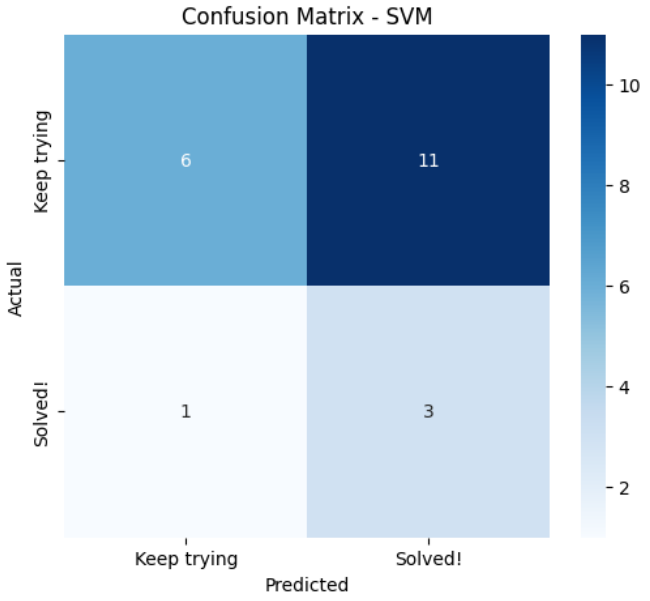

Fig. 9. Confusion Matrix For Actual vs Predicted In Towers Of Hanoi

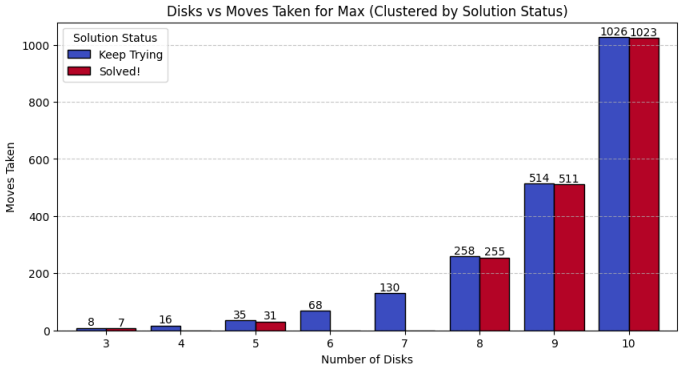

Fig. 10. Disks vs Moves Taken For Max

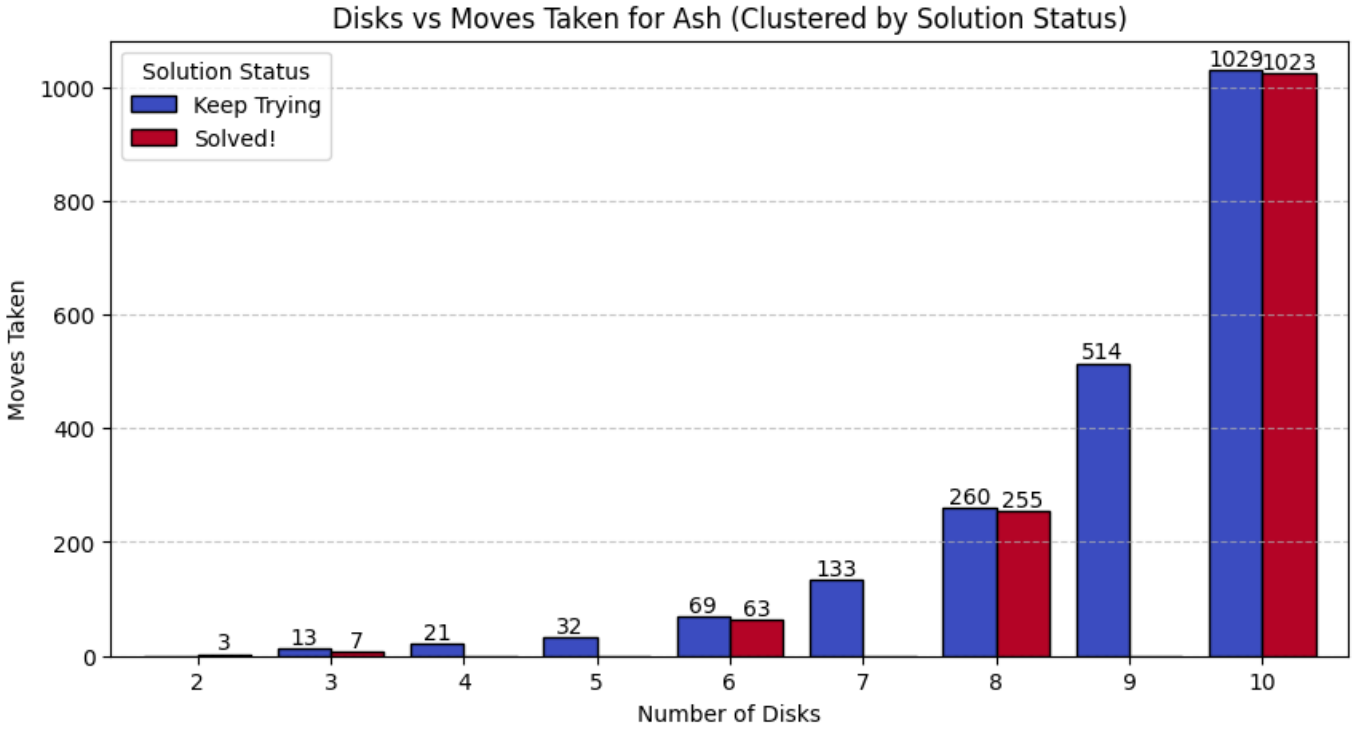

Fig. 11. Disks vs Moves Taken For Ash

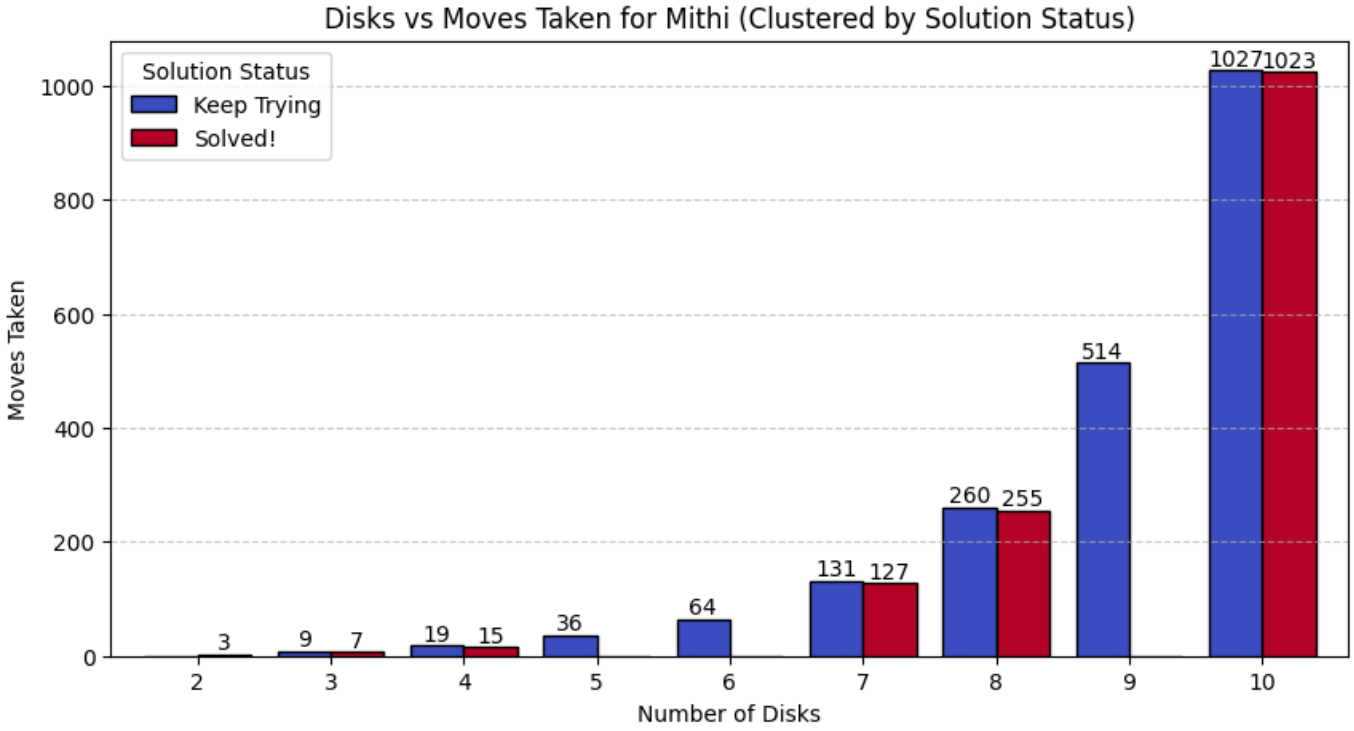

Fig. 12. Disks vs Moves Taken For Mithi

C. Josephus Problem

The Josephus problem model demonstrated high classification accuracy of 94% using Random Forest Regression as indicated by the evaluation metrics. The classification report reveals strong precision and recall for predicting wrong moves, especially for classes with higher support counts. However, certain classes such as those representing fewer wrong moves exhibited lower predictive precision due to imbalanced data distribution. Predicted results for different players showcased variability in the number of wrong moves across rounds, highlighting the inherent complexity and randomness of the problem. Despite occasional misclassifications, the model provides valuable insights into learner behavior and areas prone to errors in the Josephus problem.

TABLE IV
CLASSIFICATION REPORT SUMMARY FOR JOSEPHUS PROBLEM USING RANDOM FOREST REGRESSION

Metric	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.94	31
Macro Avg	0.73	0.77	0.75	31
Weighted Avg	0.91	0.94	0.92	31

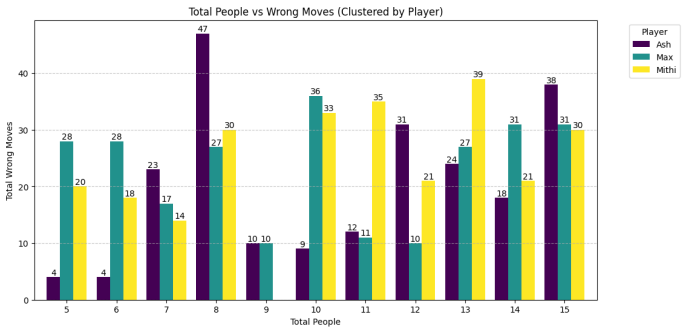

Fig. 13. Total People vs Wrong Moves For Every Player(Clustered) In Josephus Problem

Fig. 14. Total People vs Wrong Moves For Max In Josephus Problem

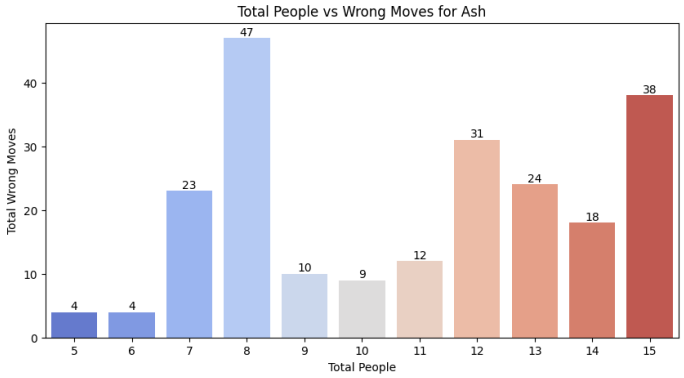

Fig. 15. Total People vs Wrong Moves For Ash In Josephus Problem

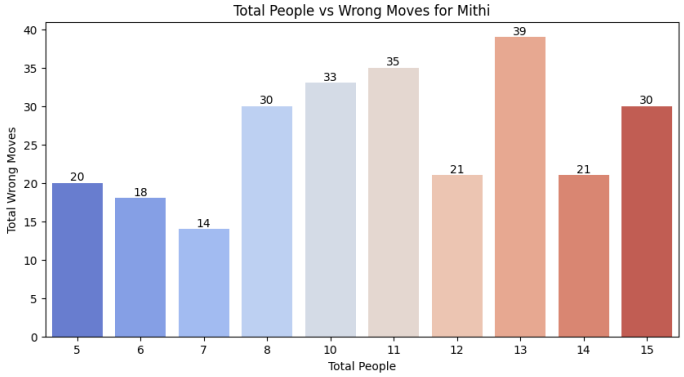

Fig. 16. Total People vs Wrong Moves For Mithi In Josephus Problem

D. N-Queens

The N-Queens problem model achieved a moderate classification accuracy of 56% using Decision Tree Classifier effectively distinguishing between "Correct placement" and "Invalid placement" scenarios to some extent. The classification report shows balanced yet slightly lower precision and recall scores, especially for the class representing invalid placements, which is likely due to data imbalance. The model demonstrates reliable predictions for smaller board sizes, though the variability increases as complexity rises. Player-specific analysis indicates that most predicted outcomes align with the actual queen placements, offering useful feedback, although there is room for improvement to enhance model consistency across larger board configurations.

TABLE V
CLASSIFICATION REPORT FOR N-QUEENS PROBLEM USING DECISION TREE CLASSIFIER

Class	Precision	Recall	F1-Score	Support
0 (Correct Placement)	0.70	0.59	0.64	27
1 (Invalid Placement)	0.39	0.50	0.44	14
Accuracy	-	-	0.56	41
Macro Avg	0.54	0.55	0.54	41
Weighted Avg	0.59	0.56	0.57	41

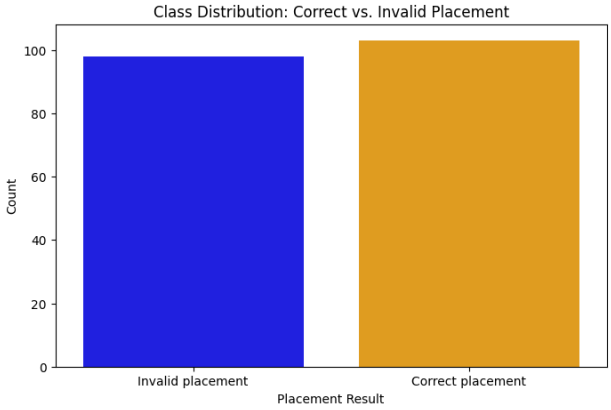

Fig. 17. Correct vs Invalid Placement In N-Queens

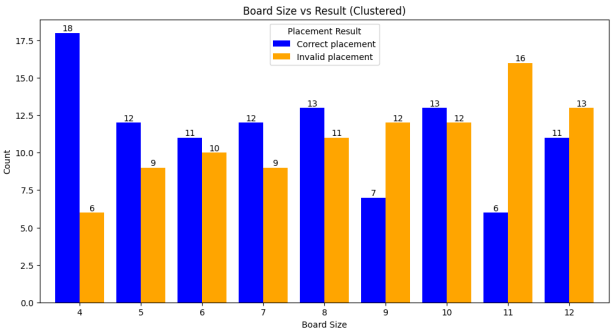

Fig. 18. Board Size vs Result In N-Queens

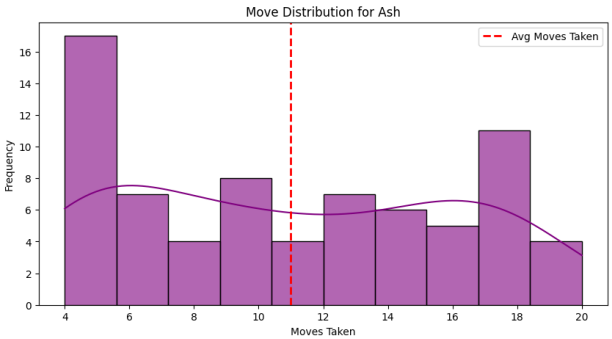

Fig. 19. Move Distribution For Ash In N-Queens

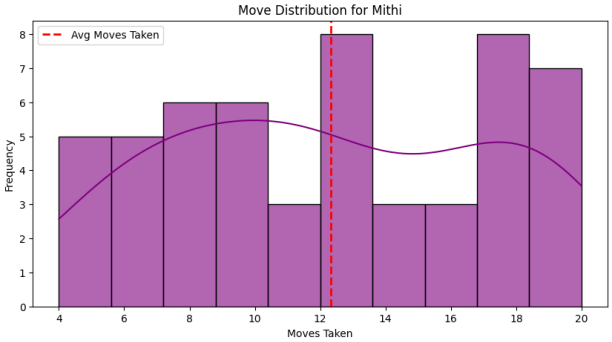

Fig. 20. Move Distribution For Mithi In N-Queens

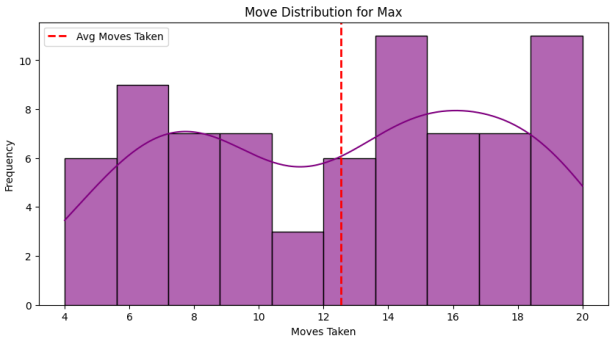

Fig. 21. Move Distribution For Max In N-Queens

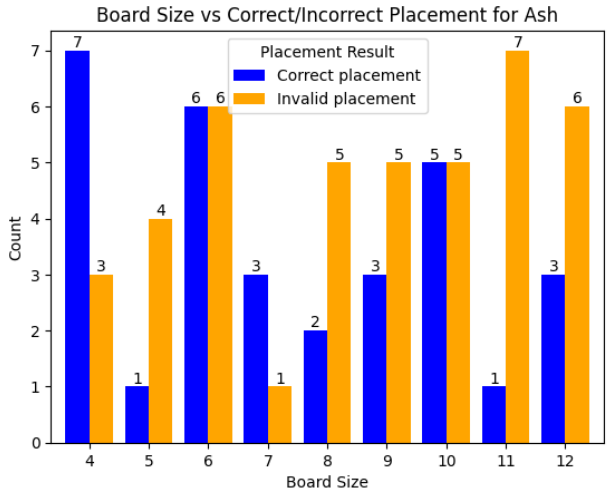

Fig. 22. Board Size vs Result For Ash In N-Queens

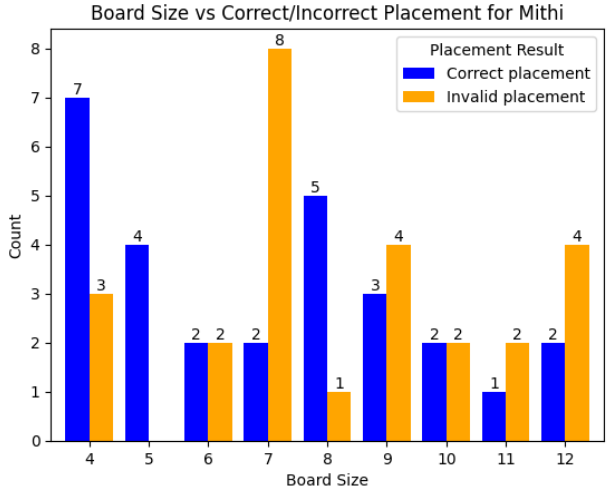

Fig. 23. Board Size vs Result For Mithi In N-Queens

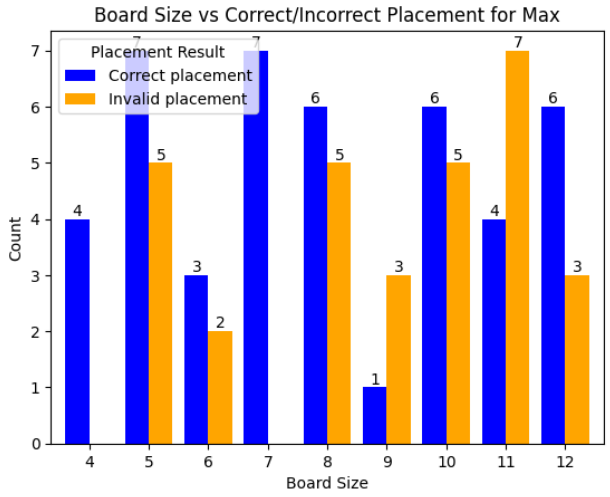

Fig. 24. Board Size vs Result For Max In N-Queens

VI. Conclusion

The development of the Animation Game Engine for Interactive Presentation of Educational Media represents a substantial contribution to the advancement of educational technology. By integrating animation, interactivity, and data-driven feedback into a unified platform, the engine addresses key challenges in modern education, particularly in conveying complex concepts in mathematics, science, and computing. The current implementation has demonstrated effectiveness in delivering dynamic and engaging educational content. Testing confirms the engine's ability to render smooth keyframe animations, capture user interactions, and provide personalized feedback based on learner performance. Furthermore, the engine's user-friendly interface ensures that educators with minimal technical expertise can create interactive presentations, fulfilling the project's core objectives.

However, there remain significant opportunities for future enhancements. One primary area for future development involves expanding the interactivity within the engine. Planned features include additional interaction types such as drag-and-drop capabilities, annotation tools, and collaborative learning environments where multiple users can interact simultaneously within the same presentation. These enhancements will further enrich the learning experience by allowing students to engage actively and collaboratively.

Another focus will be on improving the adaptability of the engine. Enhancements to the keyframe parser are anticipated, allowing for the support of more complex animations and transitions. This will offer educators greater flexibility and creativity when designing lesson content, making it easier to represent intricate processes or step-by-step algorithmic solutions.

Additionally, refining the data analytics module remains a priority. Incorporating advanced machine learning techniques will allow for deeper analysis of student engagement patterns, helping predict areas where learners may need additional support. Enhanced data privacy protocols will also be implemented to align with evolving data protection standards, ensuring the security and confidentiality of user data in educational contexts.

Finally, continued optimization efforts will focus on ensuring the engine remains lightweight and resource-efficient, maintaining compatibility with basic computing infrastructure. This will uphold the engine's commitment to supporting sustainable, low-cost, and eco-friendly digital education solutions, especially in regions with limited technological resources.

Through these ongoing improvements, the animation engine is positioned to evolve into a scalable, adaptable, and impactful tool, redefining interactive education for a diverse range of learners and educators.

References

[1] A. Kusheryanto, A. Mirza, A. Syaripudin, D. D. Hutagalung, "Development of the Story of Life: A Narrative and Educational Game Using the Godot Engine for Android", 2024.

[2] A. Kerren, "Visualizations and Animations in Learning Systems", 2012.

[3] A. Manzur, G. Marques, "Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0", Publisher, 2018.

[4] A. Bandura, J. Sweller, "The Impact of Game-Based Learning on Student Motivation and Engagement", 2020.

[5] D. Carneiro, M. Carvalho, "Algorithms Through Games. Teaching Data Structures and Lecture Notes in Networks and Systems: Methodologies and Intelligent Systems for Technology Enhanced Learning", 12th International Conference, pp. 3-12, 2022.

[6] Y. Cakiroglu, M. Ergun, "The Role of Interactive Animations in Enhancing Problem-Solving Skills", 2016.

[7] G. Kogan, H. Chassidim, I. Rabaev, "The efficacy of animation and visualization in teaching data structures: a case study", 2024.

[8] H. W. Robbins, S. C. Gutekunst, D. B. Shmoys, D. P. Williamson, "GILP: An Interactive Tool for Visualizing the Simplex Algorithm", 2023.

[9] K. P. K. Jeetha, "Animation for Learning: Enhancement of Learning through Animation: A Review of Literature", *International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, no. 1, pp. X-X, Jan. 2021.

[10] J. T. Stasko, "Using student-built algorithm animations as learning aids", 1997.

[11] K. Perlin, Z. He, K. Rosenberg, "Chalktalk: A Visualization and Communication Language – As a Tool in the Domain of Computer Science Education", 2018.

[12] L. Végh, V. Stoffová, "Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings", 2017.

[13] M. Dhule, "Beginning Game Development with Godot", Publisher, 2022.

[14] M. Ranaweera, Q. H. Mahmoud, "Deep Reinforcement Learning with Godot Game Engine", 2024.

[15] M. Krajčovič, G. Gabajová, B. Furmannová, V. Vavrík, M. Gašo, M. Matys, "A Case Study of Educational Games in Virtual Reality as a Teaching Method of Lean Management", 2021.

[16] M. Paredes-Velasco, J. Á. Velázquez-Iturbide, M. Gómez-Ríos, "Augmented reality with algorithm animation and their effect on students' emotions", 2022.

[17] S. Goel, V. Varshney, S. Dikshant, A. Sharma, S. Johri, "A Review of The Algorithm Visualization Field", 2023.

[18] V. J. Shute, F. Ke, "Games, Learning, and Assessment: Progressive Design of Games to Facilitate Learning and Assessment", *Assessment in Game-Based Learning*, Springer, pp. 43-58, 2012.

[19] S. Hansen, N. H. Narayanan, "On the Role of Animated Analogies in Algorithm Visualizations", 2000.

[20] S. Hansen, N. H. Narayanan, M. Hegarty, "Designing Educationally Effective Algorithm Visualizations", 2002.

[21] S. Su, E. Zhang, P. Denny, N. Giacaman, "A Game-Based Approach for Teaching Algorithms and Data Structures Using Visualizations", *SIGCSE '21: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pp. 1128-1134, 2021.

[22] A. Wilson, L. Martin, ”Applications of Interactive Media in Education: Study on Adaptive Learning”, 2019.
[23] B. Wu, B. Richards, ”An Overview of Gamification in E-Learning and Its Effects on Motivation and Learning”, *Computers in Human Behavior Reports*, vol. 3, p. 100086, 2021.