ANIMATION GAME ENGINE FOR INTERACTIVE PRESENTATION OF EDUCATIONAL MEDIA

PHASE II REPORT

Submitted By

ASHWIN R **3122215001014**

MATLI MITHILESH REDDY **3122215001050**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam - 603110

May 2025

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled **"ANIMATION GAME ENGINE FOR INTERACTIVE PRESENTATION OF EDUCATIONAL MEDIA"** is the *bonafide* work of "**ASHWIN R (3122215001014)** and **MATLI MITHILESH REDDY (3122215001050)**", who carried out the project work under my supervision.

Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

DR. T.T. MIRNALINEE
HEAD OF THE DEPARTMENT
Professor
Department of CSE
SSN College of Engineering

Kalavakkam - 603 110

Dr. R.S. MILTON
SUPERVISOR
Professor
Department of CSE
SSN College of Engineering

Kalavakkam - 603 110

Submitted for project viva-voce examination held on. . . . . .

EXTERNAL EXAMINER **INTERNAL EXAMINER**

ACKNOWLEDGEMENTS

We are deeply grateful to the Almighty for giving us the strength and knowledge to successfully complete this project. We extend our heartfelt gratitude to our guide, **Dr. R.S. MILTON**, Professor, Department of Computer Science and Engineering, for his guidance and continuous support. His invaluable contributions played a crucial role in shaping and refining our work.

We would also like to express our sincere gratitude to **Dr. T.T. MIRNALINEE**, Head of the Department of Computer Science and Engineering, for playing her part in encouraging us to undertake a project of this caliber, thereby providing us with the opportunity to implement an idea of this scale and complexity.

Special thanks to the founder, **Dr. SHIV NADAR**, Chairman of SSN Institutions, and to **Dr. S. RADHA**, Principal of SSN College of Engineering, for their pivotal roles and contributions in establishing an environment that enabled us to undertake and successfully complete the project.

Finally, we express our deepest appreciation to our parents and our peers for their unwavering moral support throughout our journey.

ASHWIN R **MATLI MITHILESH REDDY**

ABSTRACT

In teaching complex subjects like Mathematics, Science, and Computing, animation and visualization significantly enhance learners' understanding. However, traditional educational tools, including static presentations and printed materials, often fail to capture dynamic processes or foster meaningful engagement. These tools lack interactivity, adaptability, and real-time feedback, resulting in passive learning experiences. To overcome these limitations, this project introduces an interactive Animation Game Engine developed using the Godot Engine and GDScript. The engine enables educators to create visually dynamic, keyframe-based animations, coupled with interactive elements such as quizzes and manipulable variables, without requiring advanced programming skills. It allows learners to engage actively with educational content, receive real-time feedback, and follow personalized learning paths, thereby promoting deeper comprehension and retention. Moreover, the system records and analyzes user interactions, providing educators with valuable insights into student progress. Lightweight, accessible, and adaptable across disciplines, the engine revolutionizes traditional teaching methodologies by transforming static presentations into dynamic, interactive educational experiences, paving the way for a more inclusive and effective learning environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

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.

1.1 MOTIVATION

In elucidating complex concepts to learners in fields like mathematics, science, and computing, animation and visualization are essential in bridging the gap between theory and practical understanding. Research consistently highlights that learners benefit significantly from visual learning aids, especially when dealing with abstract or multi-step problems. However, most conventional educational tools fall short in terms of multimedia integration and interactivity, resulting in static and one-sided presentations that often lead to disengagement and surface-level understanding.

A critical issue with many of these tools is their technical complexity. While some platforms enable the creation of animated educational content, they typically require advanced programming knowledge, deterring educators who lack such skills from adopting them. The lack of meaningful interactivity in the resulting content further limits their effectiveness, as students are relegated to passive observation rather than active engagement. This not only reduces the

depth of comprehension but also hinders the ability to provide immediate, personalized feedback—a key factor in effective learning.

This project was motivated by the need to address these limitations. By leveraging the open-source Godot Engine and its scripting language GDScript, the animation engine simplifies the process of developing interactive, animated presentations. Educators can easily incorporate elements like quizzes, real-time feedback, adaptive content, and data-driven insights without requiring extensive coding experience. Keyframe-based animation support allows educators to animate sequences with minimal effort, while user interactions are tracked and analyzed to provide valuable feedback on learner progress.

Furthermore, the motivation extends to inclusivity and accessibility. The engine's lightweight design ensures compatibility with a wide range of hardware, enabling its use even in resource-constrained environments. This democratizes access to high-quality educational technology, particularly benefiting schools and institutions in developing regions.

By removing technical barriers and enhancing the interactive capabilities of educational media, this project aims to make learning more engaging, adaptive, and effective, ultimately empowering both educators and learners to achieve better educational outcomes.

1.2 BACKGROUND

In today's fast-evolving educational landscape, traditional teaching methods often prove insufficient for effectively conveying the complexities of subjects like mathematics, science, and computing. These subjects inherently involve dynamic processes, abstract relationships, recursive procedures, and intricate transformations that cannot be easily communicated through static instructional materials. Conventional presentation tools, such as slide-based software, printed textbooks, or static diagrams, typically lack the depth of animation and interactivity required to facilitate a deeper understanding of such concepts. Consequently, the learning experience remains predominantly passive, with students limited to consuming information rather than actively engaging with it. This one-directional flow of information can hinder comprehension, limit motivation, and reduce the long-term retention of knowledge, particularly in disciplines that benefit from step-by-step visualization or exploratory problem-solving activities.

To address these limitations, this project introduces a robust and versatile Animation Game Engine, specifically tailored for educational media, developed using the Godot Engine and GDScript. This open-source, lightweight platform provides an accessible, user-friendly environment for educators to create highly interactive, dynamic presentations without requiring extensive coding experience. By supporting keyframe-based animations, educators can visually illustrate complex concepts, algorithms, or scientific processes in an intuitive manner. Additionally, the engine integrates features such as quizzes, drag-and-drop activities, variable manipulation, and real-time feedback, ensuring that learners remain engaged and can interact with the material in meaningful ways.

One of the engine's significant advantages is its built-in ability to track, record, and analyze user interactions throughout the learning process. By capturing data on quiz responses, engagement patterns, user inputs, and overall interaction behavior, educators are equipped with valuable insights into individual learner progress. This data-driven approach allows educators to adapt their teaching strategies based on real-time feedback, identifying learning gaps and providing targeted reinforcement where necessary. The inclusion of immediate, personalized feedback not only supports students in correcting mistakes as they occur but also fosters a more interactive and supportive learning environment.

Furthermore, the engine's design prioritizes performance efficiency and cross-platform compatibility. Its lightweight architecture ensures smooth operation across a wide range of devices, from high-end computers to resource-constrained classroom setups, thereby democratizing access to high-quality interactive educational tools. By removing technical and financial barriers, this animation engine promotes inclusivity and ensures that educational institutions, regardless of their technological capabilities, can enhance the quality of their instructional methods.

In summary, this project bridges the gap between traditional education and modern technological capabilities. By integrating multimedia-rich animations, interactivity, and adaptive learning mechanisms into a single cohesive platform, the engine redefines how complex subjects can be taught and learned. It aligns with contemporary educational trends that emphasize engagement, personalization, and data-informed teaching, thereby empowering educators to deliver more effective, engaging, and accessible learning experiences.

1.3 PROBLEM DEFINITION

The goal is to develop a flexible and interactive animation engine for educational media using the Godot Engine and GDScript. Traditional educational tools lack interactivity, adaptability, and real-time feedback, leading to passive learning experiences. Static materials struggle to convey dynamic processes, making it difficult for learners to grasp abstract concepts.

To address these challenges, the animation engine will:

- Provide keyframe-based animations for smooth visual representation.
- Offer an intuitive scripting interface for easy content creation.
- Integrate interactive features like quizzes and variable manipulation.
- Enable real-time user tracking and personalized feedback.
- Ensure lightweight performance for broad accessibility.

1.4 ORGANIZATION OF THE REPORT

1.4.1 Introduction

This section provides an overview of the project, outlining the motivation, objectives, and the need for interactive and dynamic tools to teach complex concepts effectively. It highlights the role of animation and visualization in enhancing the educational experience.

1.4.2 Background and Literature Review

This section examines existing tools and research in the field of educational animation platforms, identifying the limitations of traditional static teaching methods. It also reviews related literature supporting the use of multimedia, interactivity, and real-time feedback in education, establishing the research gap addressed by this project.

1.4.3 Problem Statement and Project Justification

This section clearly defines the challenges faced by current educational tools, such as lack of interactivity, adaptability, and personalized feedback. It justifies the need for a Godot-based animation engine to provide dynamic, interactive learning environments tailored to diverse educational needs.

1.4.4 System Design and Architecture

This section details the architecture of the animation game engine, describing its modular design. It includes the core components such as the Godot Engine project setup, scene hierarchy, GDScript integration, keyframe-based animation system, algorithm modules (e.g., Towers of Hanoi, N-Queens), and data analysis module. It also illustrates the overall workflow through a structured architecture diagram.

1.4.5 Implementation

This section presents the technical implementation of the engine using Godot Engine and GDScript. It discusses scene creation, scripting techniques, algorithm integration (sorting, pathfinding, problem-solving algorithms), and interactive features. Sample implementations, including visualizations of algorithms like A* pathfinding, Towers of Hanoi, N-Queens, and Dutch National Flag problem, are highlighted.

1.4.6 Results and Analysis

Here, the report presents experimental results of the animation engine, showcasing visual outputs of algorithm visualizations. It includes performance metrics, such as accuracy and efficiency of implemented algorithms.

1.4.7 Conclusion and Future Work

This concluding section summarizes the achievements of the project, emphasizing its impact on improving interactive learning. It identifies areas for future enhancements, such as refining animation quality, integrating IPE vector graphic parsing, improving data analysis capabilities, and extending API support for broader applications.

CHAPTER 2

LITERATURE SURVEY

2.1 PAPERS RELATED TO THE PROBLEM

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 emphasizes 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.

2.2 GAPS IDENTIFIED

Despite advancements in educational technology, there remains a significant gap in tools that combine dynamic animation, interactivity, and real-time feedback in a single, accessible platform. Traditional presentation tools often provide passive learning experiences, lacking the ability to dynamically respond to learner input

or provide personalized feedback. While algorithm visualization tools and game-based learning environments exist, many demand extensive technical expertise, making them impractical for educators without programming backgrounds.

Moreover, although sophisticated visualizations are available, they often fail to capture learner interaction data or adapt content based on real-time user behavior. This limitation restricts their ability to offer tailored feedback and adaptive learning paths, which are crucial for addressing individual learner needs and improving comprehension of complex concepts.

The research gap thus lies in the lack of a lightweight, user-friendly tool that integrates animation, interactivity, and data-driven feedback while remaining accessible to educators with minimal coding knowledge.

2.3 RESEARCH OBJECTIVES

The objective of this project is to design and develop a versatile animation game engine that addresses the identified gaps by offering a dynamic, interactive, and adaptive learning environment. Specifically, the goals are:

- To create an accessible animation engine using the Godot Engine and GDScript, enabling educators to design interactive presentations without requiring advanced programming skills.
- To support keyframe-based animations, allowing smooth and precise visualization of complex concepts such as algorithms, mathematical transformations, and scientific processes.

- To incorporate interactive elements such as quizzes, clickable objects, and variable manipulation, fostering active learner engagement.
- To implement real-time tracking and analysis of learner interactions, providing educators with valuable insights and enabling personalized feedback.
- To ensure the engine is lightweight and adaptable, capable of running on a wide range of devices, thereby democratizing access to advanced educational tools.

CHAPTER 3

PROPOSED METHODOLOGY

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.

By merging animation, interactivity, and data analytics, the proposed methodology aims to create a versatile educational tool that enhances comprehension, increases student engagement, and facilitates personalized learning experiences across various domains such as mathematics, science, and computing.

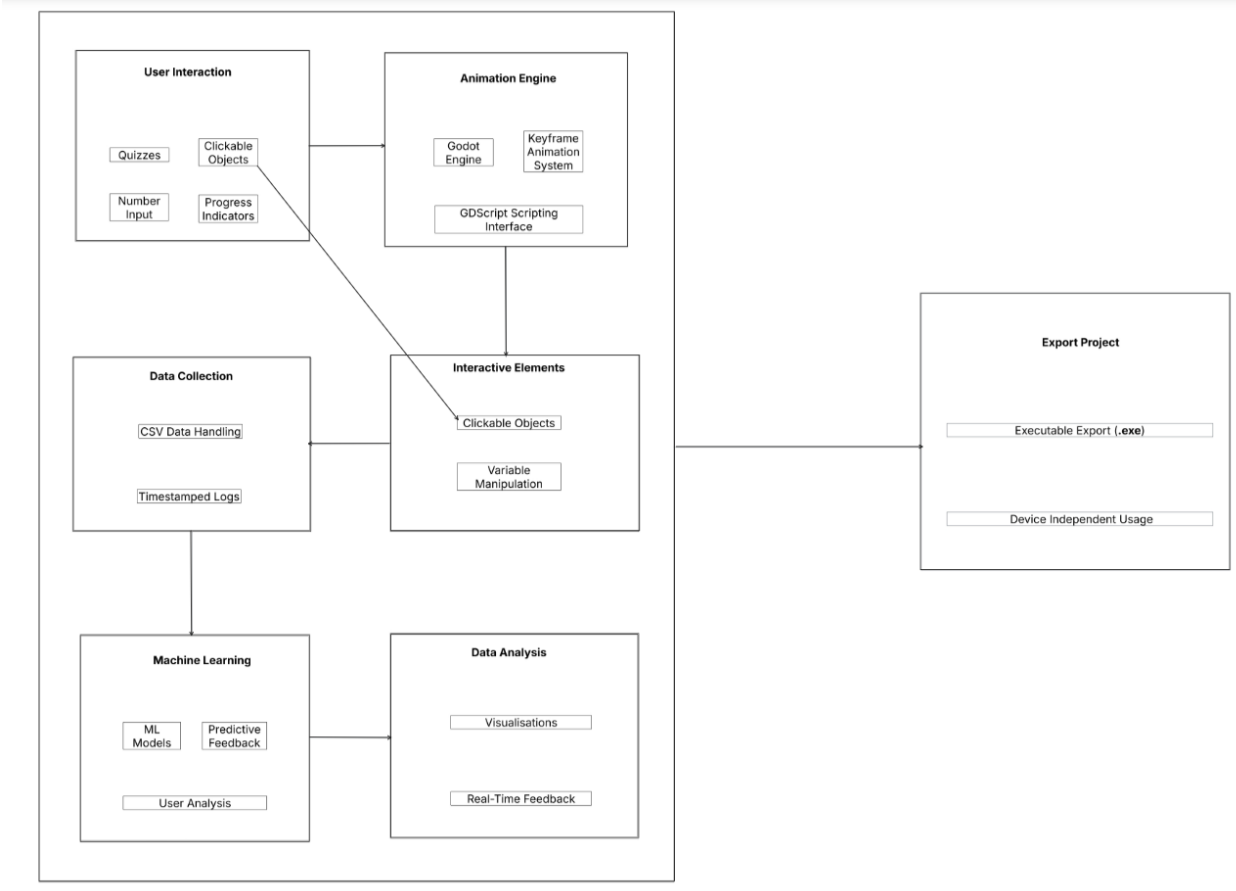

FIGURE 3.1: Architecture Diagram

3.1 ANIMATION GAME ENGINE

At the core of the proposed system lies the animation game engine, architected using the Godot Engine's powerful node-based structure. This engine acts as the backbone for developing interactive, visually appealing, and adaptable educational media. Godot's inherent modularity is leveraged to organize content into reusable, manageable nodes, each handling specific aspects like rendering animations, processing user inputs, managing quizzes, and capturing engagement data.

One of the key features of the engine is its comprehensive suite of animation primitives, which allows educators to design smooth and dynamic keyframe-based animations. These animations can visually illustrate complex scientific processes, algorithmic steps, or mathematical transformations in a step-by-step manner, making abstract concepts easier to grasp. The keyframe system is implemented in a way that requires minimal scripting effort, allowing educators to define motion sequences, transitions, and element behaviors without in-depth programming knowledge.

In addition to animation capabilities, the engine integrates interactive components such as embedded quizzes, clickable objects, drag-and-drop activities, and variable manipulation interfaces. These features promote active learner participation, transforming the learning process from passive observation to hands-on exploration. Real-time feedback mechanisms are also built into the system, enabling immediate assessment of learner responses and reinforcing correct understanding.

The data collection module within the engine records detailed information about user interactions, including quiz results, time spent on tasks, and engagement patterns. This data is processed to generate analytics that provide insights into learner performance, facilitating personalized learning paths and adaptive content delivery.

Furthermore, the engine's lightweight design ensures that it remains efficient and responsive across a variety of hardware configurations, from high-end systems to basic classroom devices. Its open-source nature allows for continuous customization and expansion, ensuring that it can evolve to meet emerging educational needs and technological advancements.

In essence, the animation game engine forms the foundation of an innovative educational platform that blends animation, interactivity, and data-driven personalization to create an engaging and effective learning experience.

3.2 TECHNOLOGIES USED

The animation engine utilizes the following technologies:

- **Godot Engine:** An open-source, cross-platform game engine known for its flexibility and ease of use. It supports both 2D and 3D graphics rendering and offers a robust node-based development environment. Additionally, the engine supports exporting the final interactive presentation as standalone executables (`.exe`) for easy deployment across different systems without requiring the Godot environment.

- **GDScript:** Godot's high-level, Python-like scripting language that allows educators to control animation sequences, implement logic, manage user interactions, and define quizzes and feedback mechanisms without the need for complex coding.

- **CSV Data Handling:** Used for structured storage and analysis of learner interaction data. User inputs, quiz responses, and engagement patterns are recorded in CSV format, enabling efficient tracking and feedback mechanisms.

- **Machine Learning Module:** Machine learning techniques are applied to analyze the collected interaction data. Models such as decision trees and random forests are utilized to classify learner performance and provide predictive feedback based on interaction patterns, further enhancing personalized learning.

- **Export as Executable:** Godot's export capabilities allow the animation engine to be packaged and distributed as executable files (`.exe`), ensuring compatibility and ease of use across various platforms without requiring additional installations.

3.3 DETAILED DESIGN

The proposed animation game engine is designed with modularity, scalability, and maintainability as its core principles. Each component is carefully structured to ensure that educators and learners experience an interactive, visually rich, and data-driven learning environment. The system is constructed around the Godot Engine, leveraging its node-based architecture and scripting flexibility through GDScript. The design emphasizes the smooth rendering of animations, ease of content creation, real-time interactivity, and the ability to collect and analyze user interaction data for adaptive learning.

The key components of the system include:

- **Graphics Rendering:** Handled by Godot's high-performance rendering engine. It supports 2D and 3D scenes, allowing for smooth visualization of complex processes. The rendering engine ensures that animations and visual elements remain fluid and engaging, even when running on low-end hardware setups. The modular scene structure simplifies the addition of new visuals and supports dynamic scene management.

- **Scripting Interface:** GDScript is used to control the behavior of all interactive elements, including animations, quizzes, and user-driven actions. Educators can define keyframe animations, create interactive scenes, and control logic flows without deep technical knowledge. This simplifies the process of developing customized, interactive presentations tailored to specific subject matter.

- **Keyframe Animation System:** The engine incorporates Godot's `AnimationPlayer` nodes, enabling precise control over keyframe-based animations. Educators can define the positions, rotations, and properties of visual elements at specific time intervals, with the engine handling the interpolation between keyframes. This system makes it possible to illustrate dynamic concepts such as mathematical transformations, physics simulations, and algorithm behavior step-by-step.

- **Interactive Interface Module:** This module is responsible for embedding quizzes, clickable objects, user-controlled animations, and other interactive elements into the learning material. GDScript handles user input events, triggers animation changes, and processes quiz responses in real time. This hands-on approach allows students to manipulate variables, receive instant feedback, and engage actively with the content rather than passively observing.

- **Data Collection Module:** The engine tracks all user interactions—such as quiz answers, time spent on activities, and engagement patterns—and stores them in structured CSV format. This module ensures efficient data storage without affecting system performance. The recorded data forms the basis for real-time feedback to learners and helps educators analyze student progress over time.

- **Machine Learning Integration:** The collected interaction data is further processed using machine learning models such as decision trees, random forests, and regressors. These models classify learner performance, predict areas of difficulty, and recommend personalized feedback. The use of

machine learning enables the system to adapt dynamically to each learner's needs, offering tailored learning paths.

- **Executable Export System:** One of the key features of the engine is its ability to export the entire presentation as a standalone executable (`.exe` or other platform-specific formats) using Godot's export templates. This ensures that the interactive educational content can be deployed and used independently, without the need for additional installations or runtime environments, making the engine accessible in varied classroom settings.

The interaction between these modules is illustrated through a well-structured workflow:

1. Educators create the scenes using Godot's editor, define animations via the AnimationPlayer nodes, and script interactivity with GDScript.
2. Keyframe animations are parsed and managed by the engine to ensure smooth transitions and dynamic visualizations.
3. Interactive elements, such as quizzes and clickable objects, are integrated through predefined and user functions.
4. During student interaction, the engine records their responses, engagement levels, and behavior patterns.
5. The collected data is stored in CSV format, which is then analyzed to assess learning progress.
6. Machine learning models process the data, classify learner performance, and enable the system to offer real-time personalized feedback.

7. The complete interactive presentation is exported as an executable file, making it device-independent and easy to distribute.

3.4 CODE 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.

3.4.1 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.

```
// Sample GDScript for initialization
func _ready():
    load_resources()
    setup_ui()
    initialize_variables()
```

3.4.2 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.

```
// Sample GDScript to play animation
func play_animation(anim_name):
    $AnimationPlayer.play(anim_name)
```

3.4.3 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.

```
// Sample GDScript for quiz response handling
func _on_QuizButton_pressed(answer):
    if answer == correct_answer:
        show_feedback("Correct!")
    else:
        show_feedback("Try Again!")
```

3.4.4 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.

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

3.4.5 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.

3.4.6 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).

3.5 OUTPUT

The below output screenshots illustrate the visualizations of algorithms such as A-Star pathfinding, Dutch National Flag problem, Towers of Hanoi, Insertion Sort, Josephus Problem, N-Queens problem, Tromino Tiling, and Cellular Automaton. Each algorithm is animated dynamically using Godot's keyframe-based animation system and allows for learner interaction through real-time inputs.

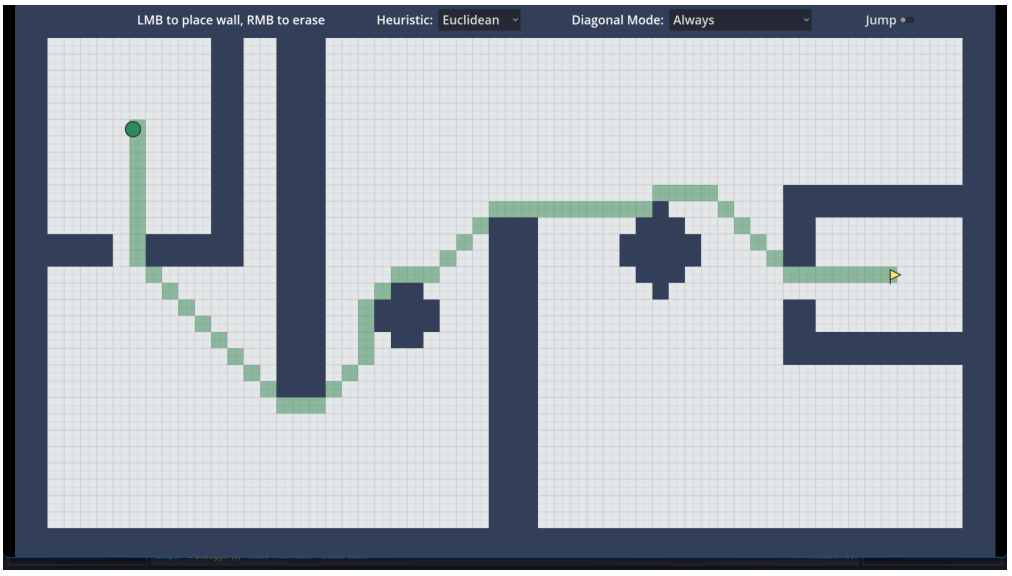

FIGURE 3.2: A-Star Algorithm Implementation

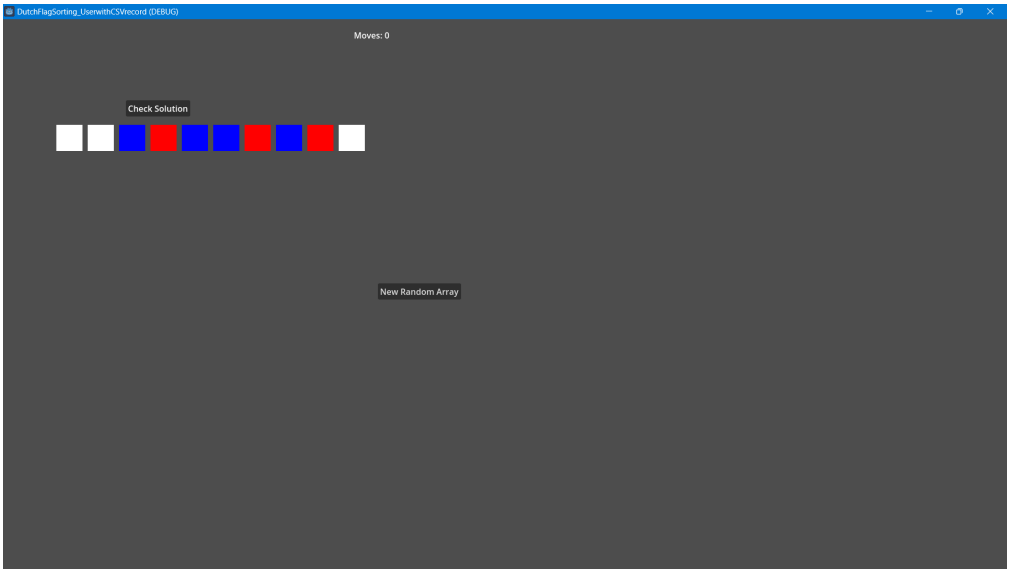

FIGURE 3.3: Dutch National Flag Algorithm Implementation

FIGURE 3.4: Towers of Hanoi Implementation

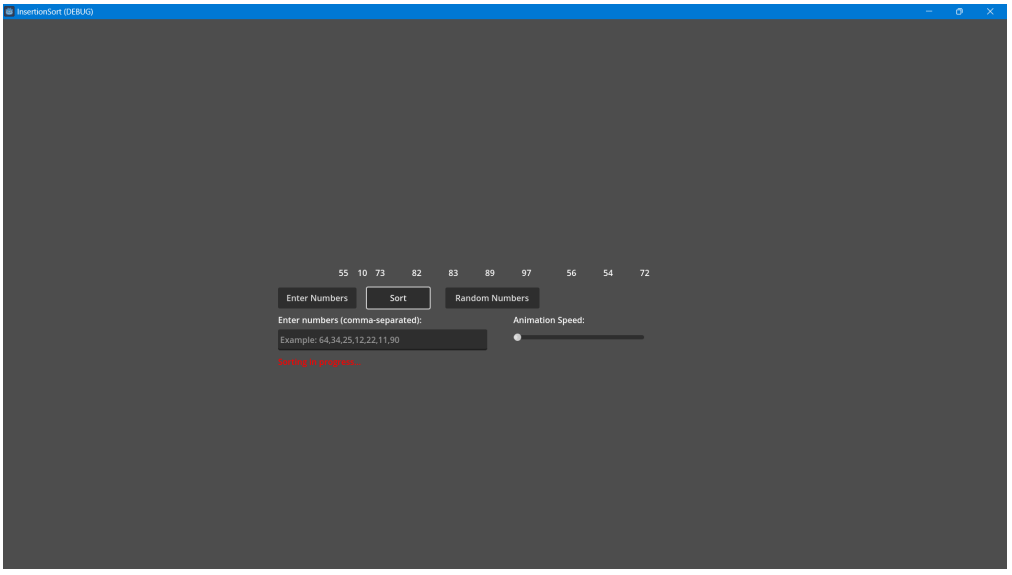

FIGURE 3.5: Insertion Sort Algorithm Implementation

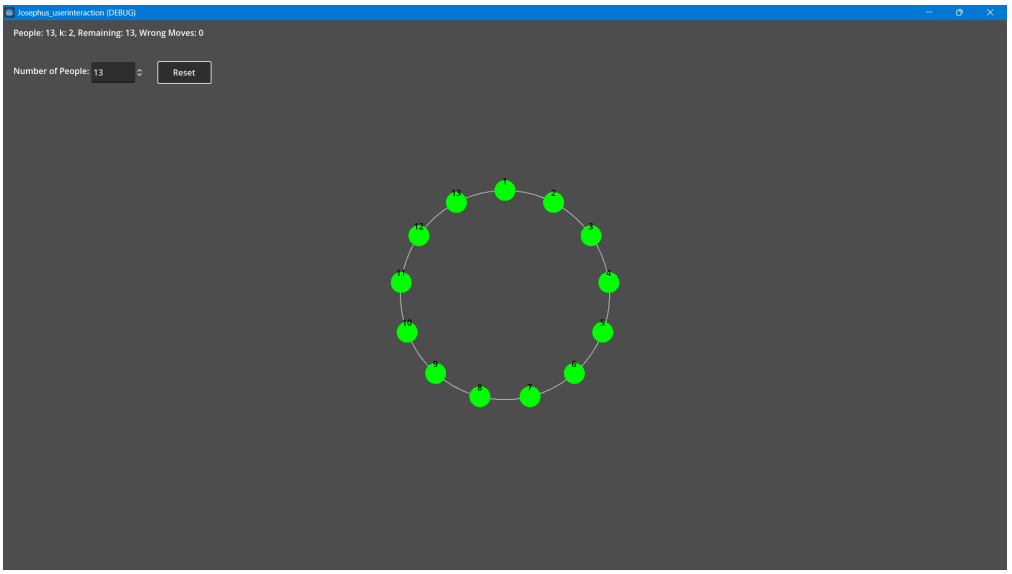

FIGURE 3.6: Josephus Problem Implementation

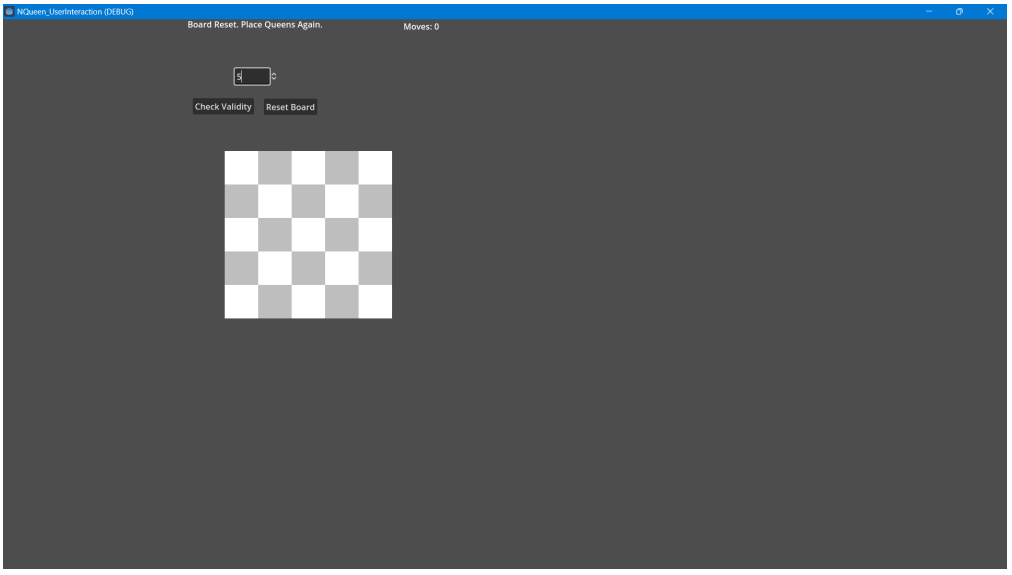

FIGURE 3.7: N-Queens Problem Implementation

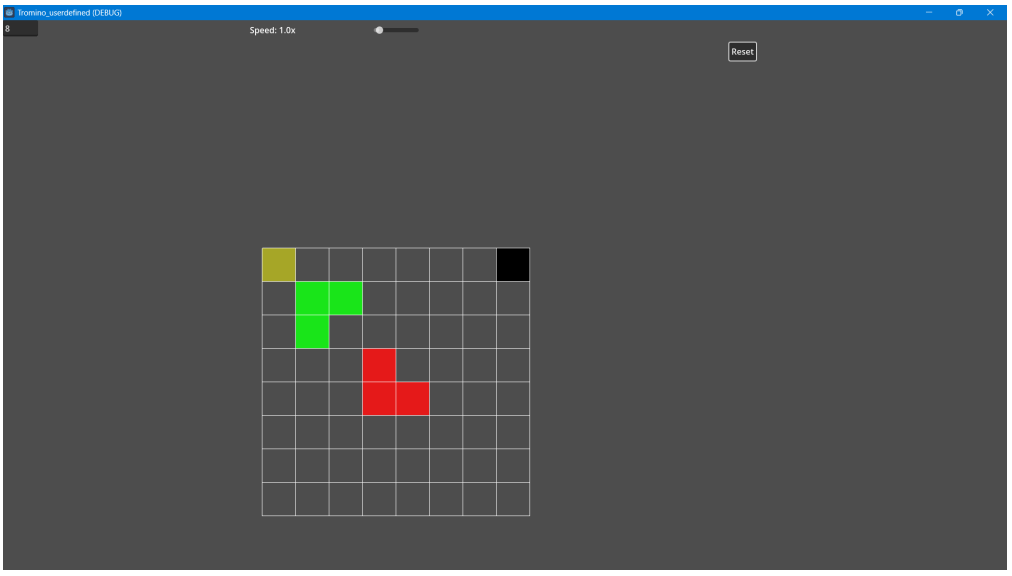

FIGURE 3.8: Tromino Tiling Problem Implementation

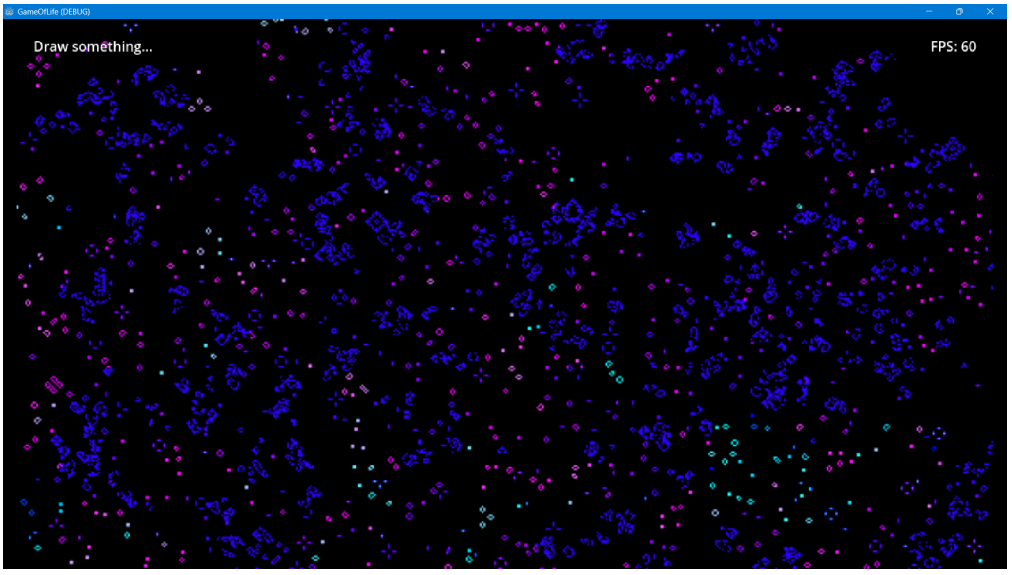

FIGURE 3.9: Cellular Automaton Implementation

CHAPTER 4

EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

4.1 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.

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

TABLE 4.1: Performance Metrics for Dutch National Flag Algorithm using Random Forest Regressor

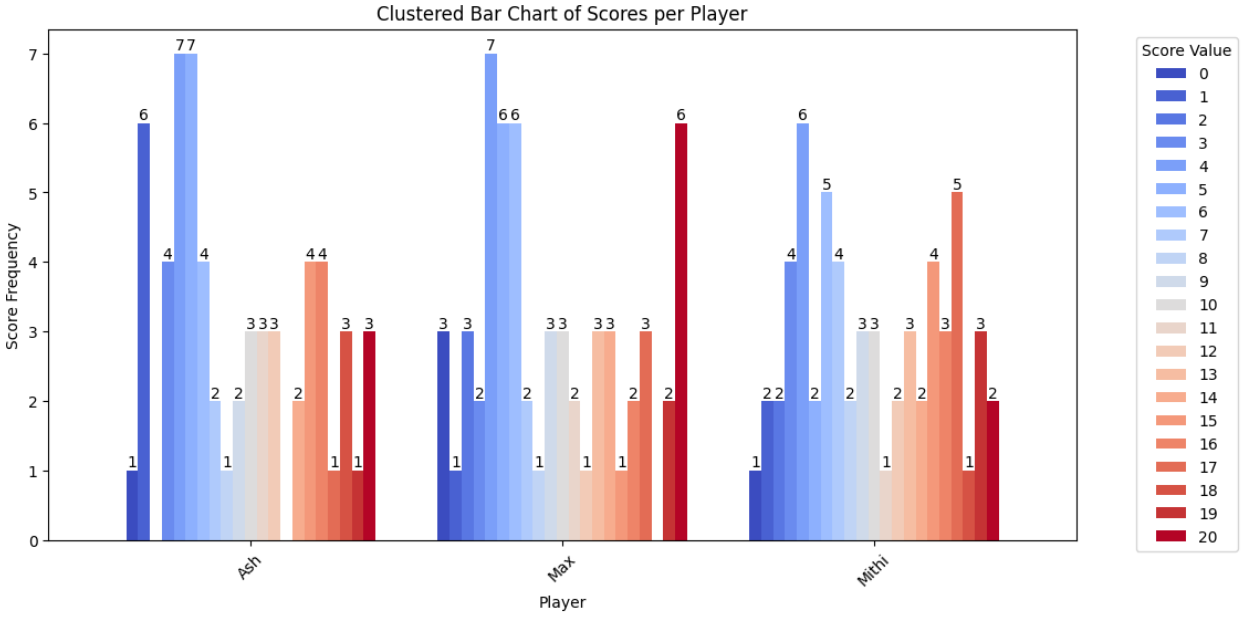

FIGURE 4.1: Clustered Bar Chart Of Scores Per Player In DNF

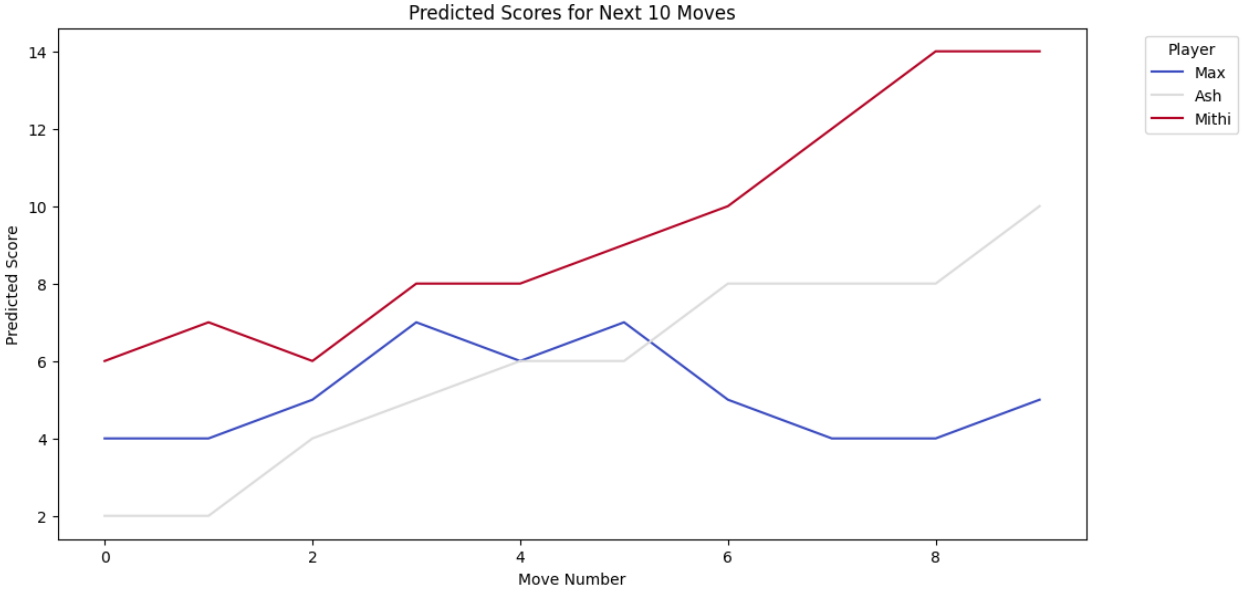

FIGURE 4.2: Predicted Scores For Next 10 Moves(dynamic) In DNF

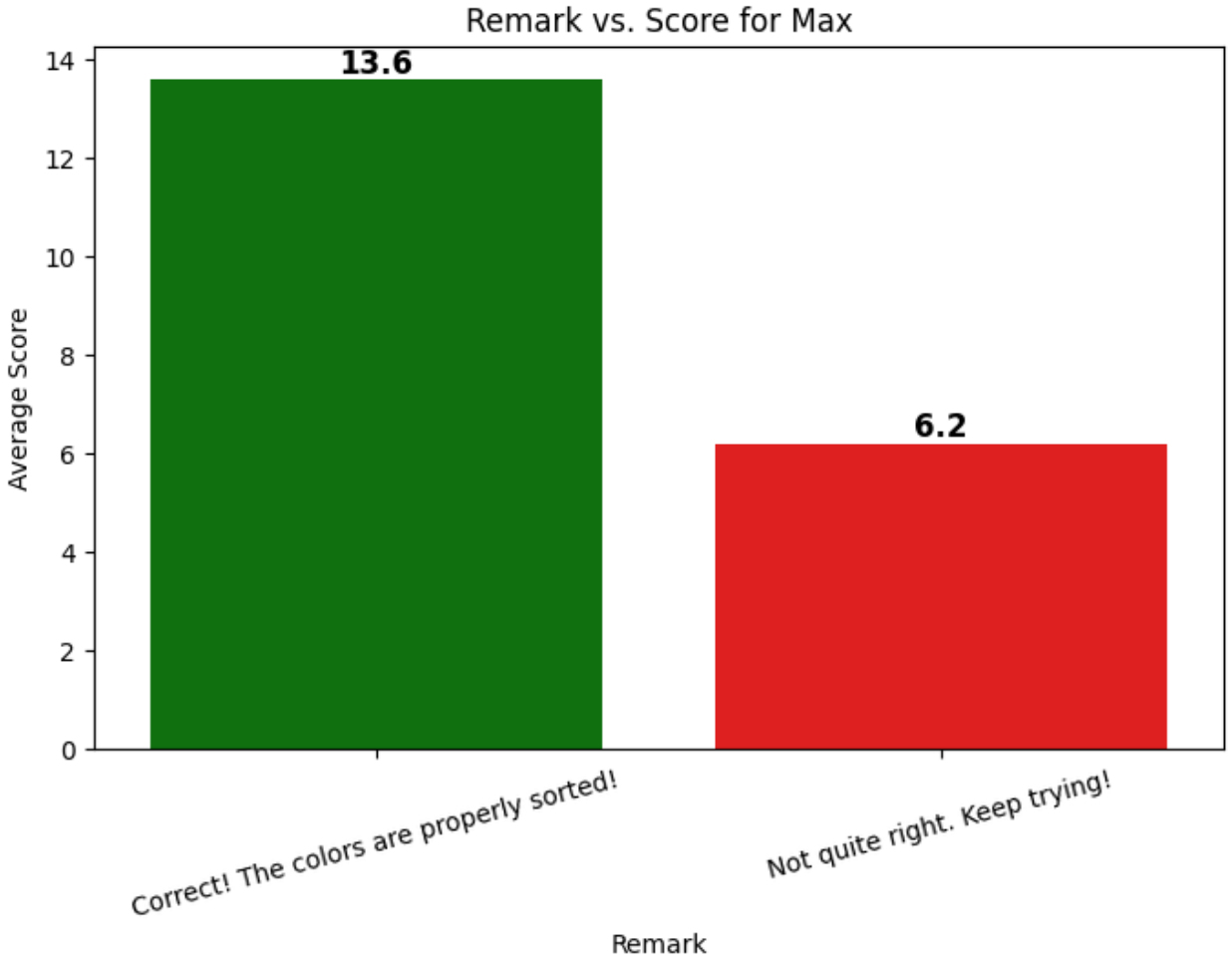

FIGURE 4.3: Remark vs Score For Max In DNF

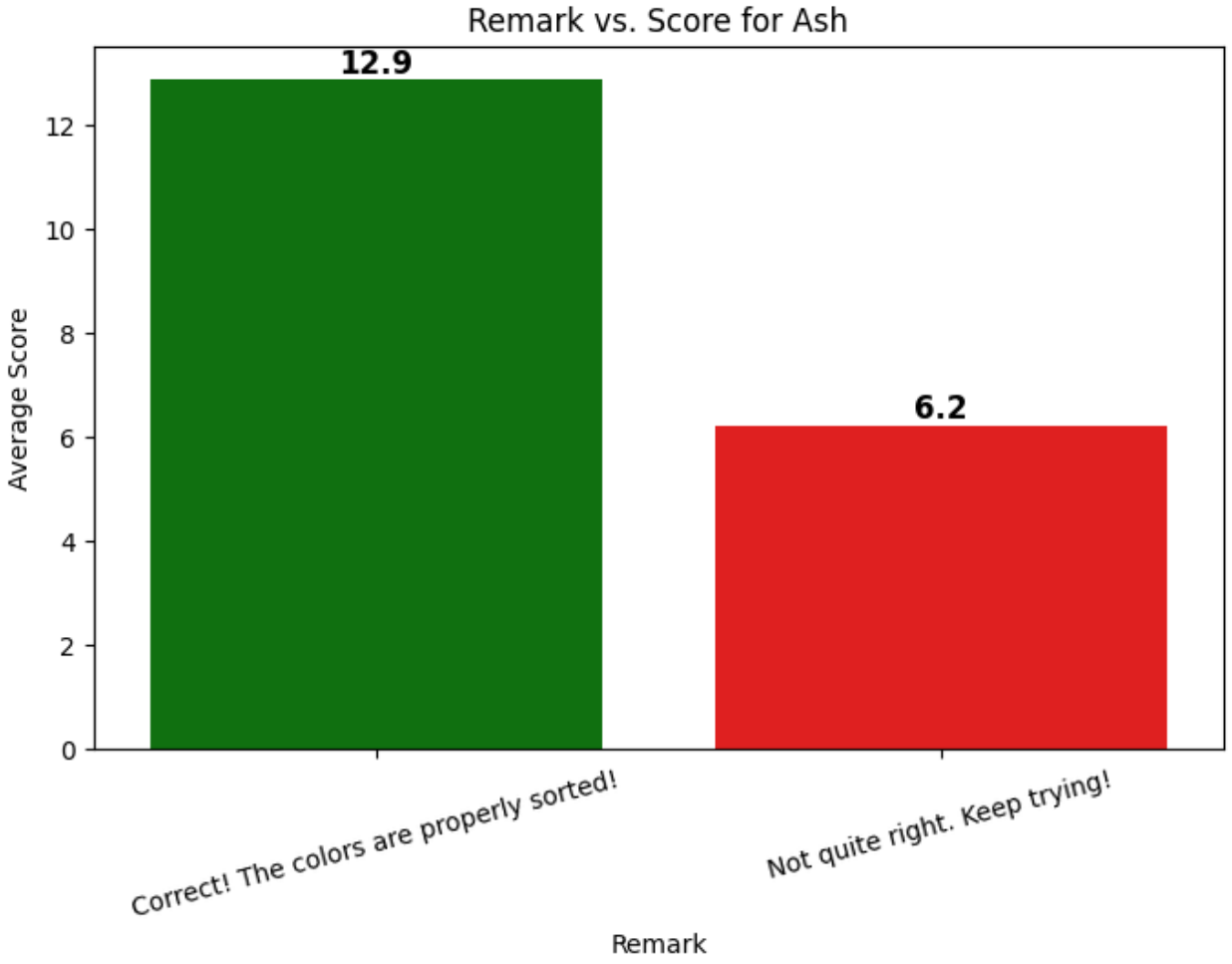

FIGURE 4.4: Remark vs Score For Ash In DNF

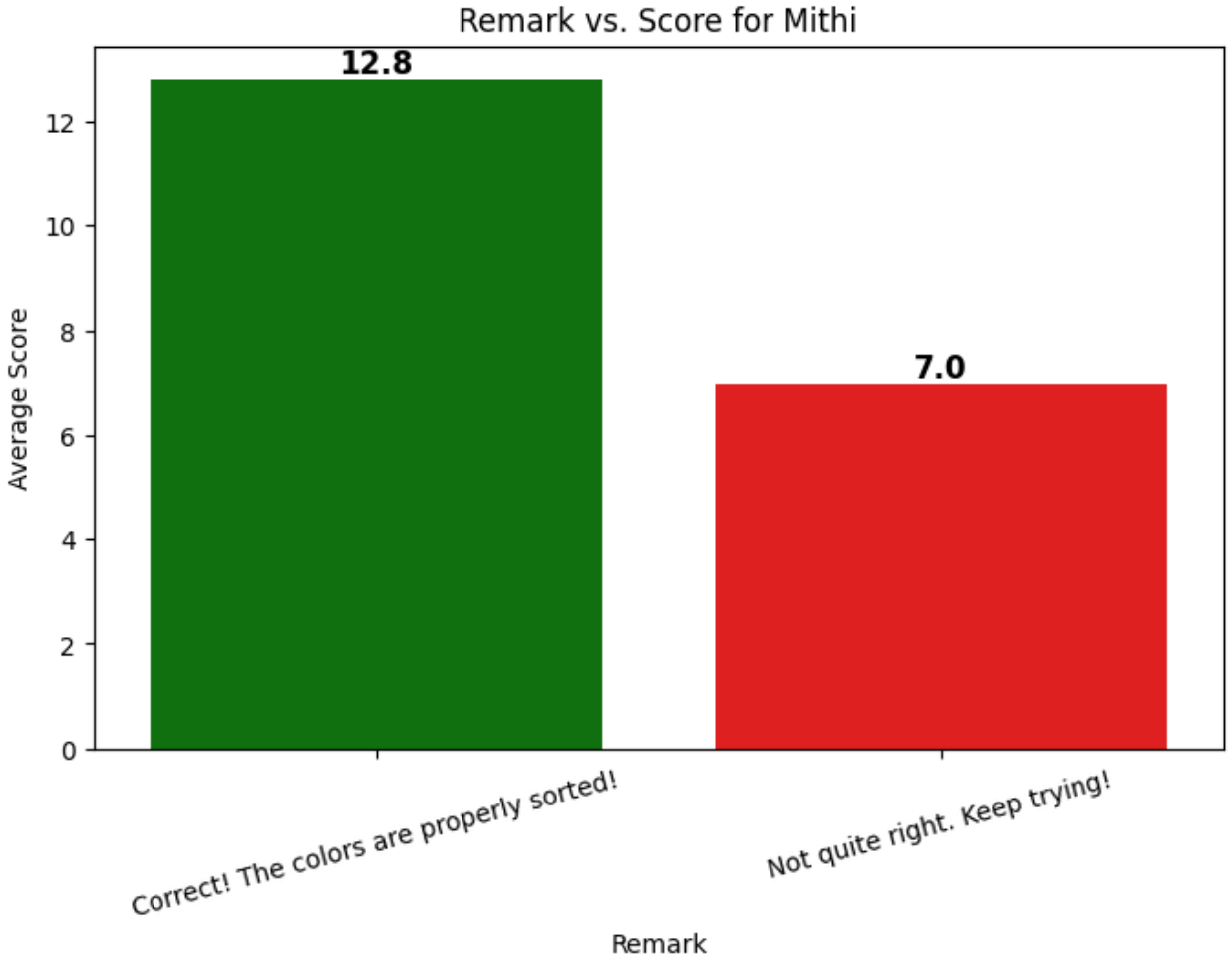

FIGURE 4.5: Remark vs Score For Mithi In DNF

4.2 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.

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 4.2: Classification Report for Tower of Hanoi using Decision Tree 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

TABLE 4.3: Classification Report for Tower of Hanoi using SVM Classifier

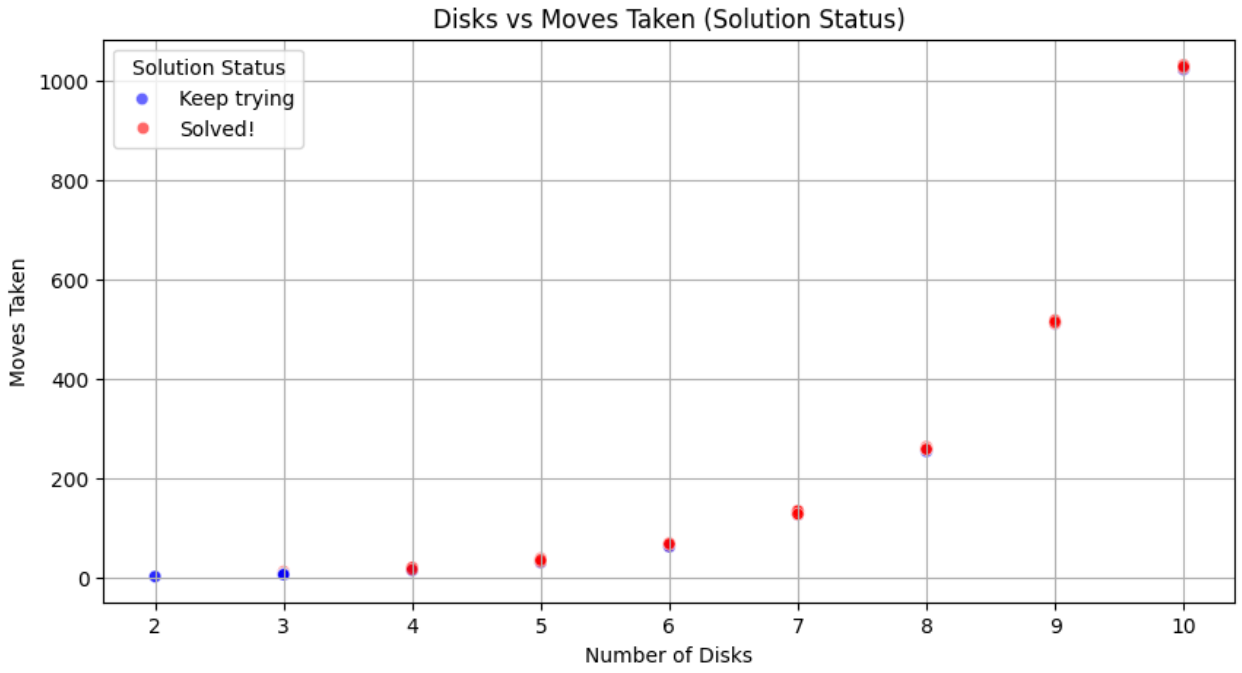

FIGURE 4.6: Disks vs Moves Taken In Towers Of Hanoi

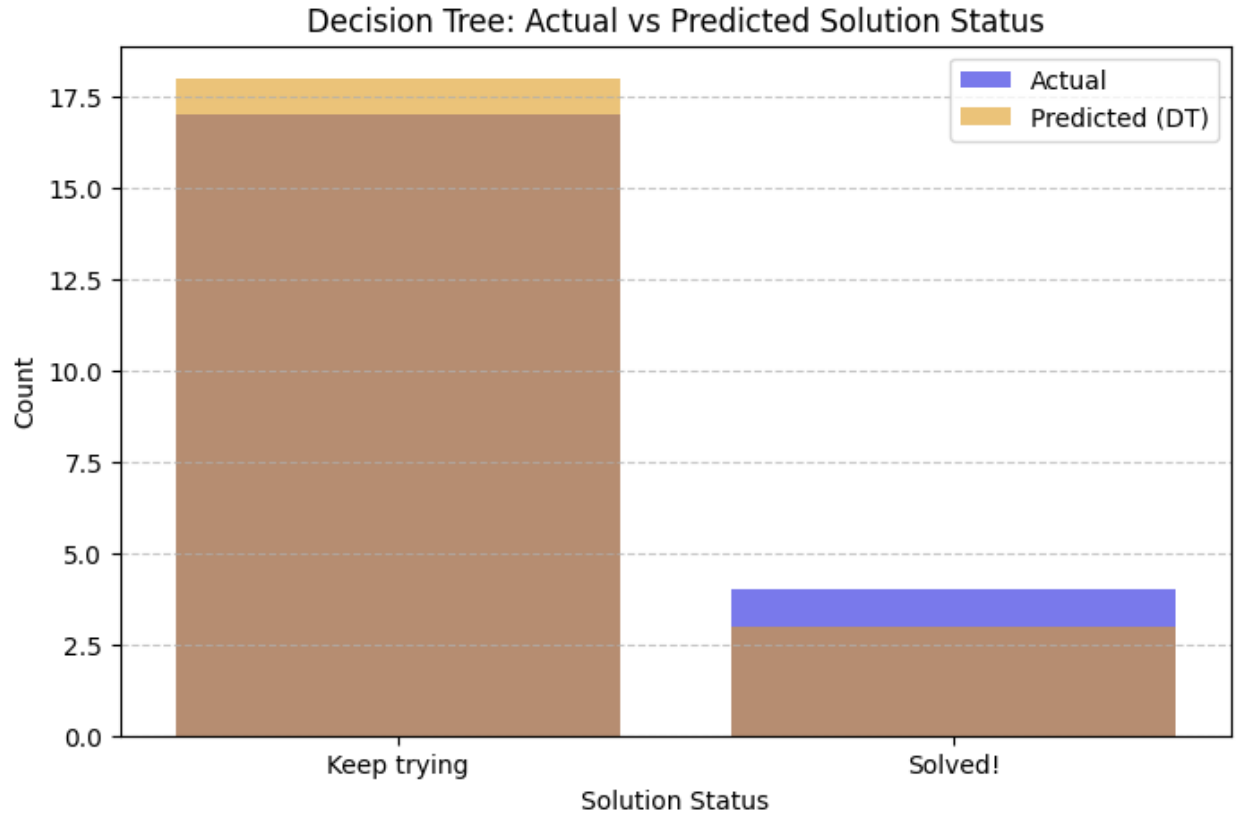

FIGURE 4.7: Actual vs Predicted Solution Status In Towers Of Hanoi

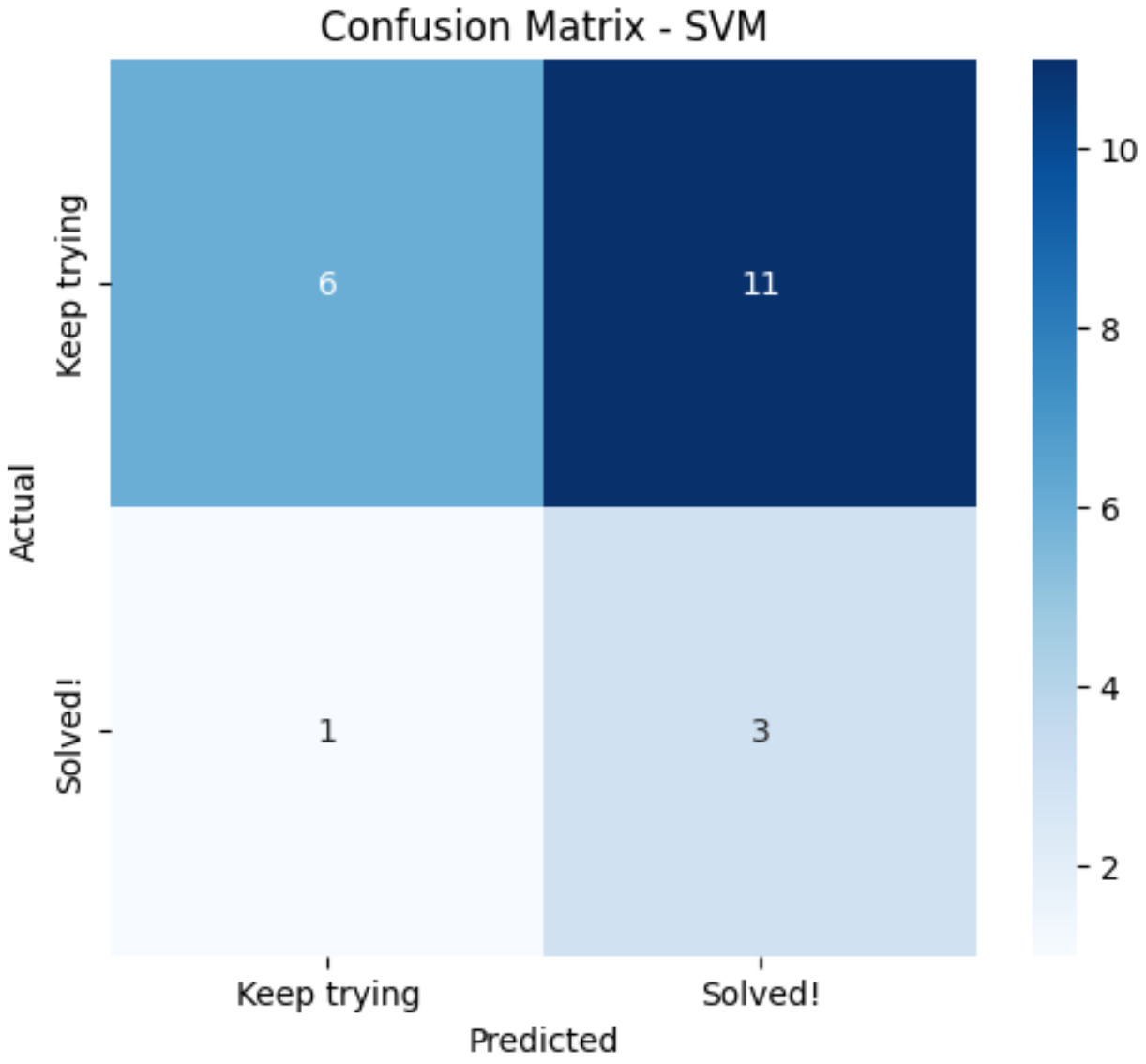

FIGURE 4.8: Confusion Matrix For Actual vs Predicted In Towers Of Hanoi

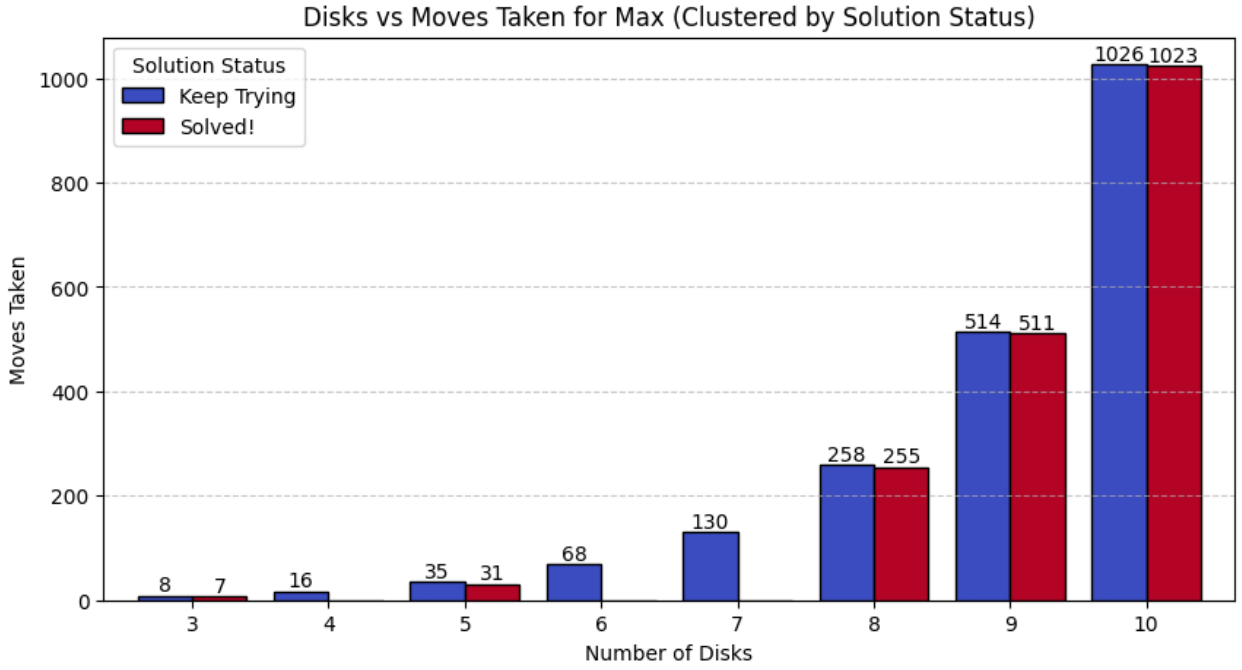

FIGURE 4.9: Disks vs Moves Taken For Max

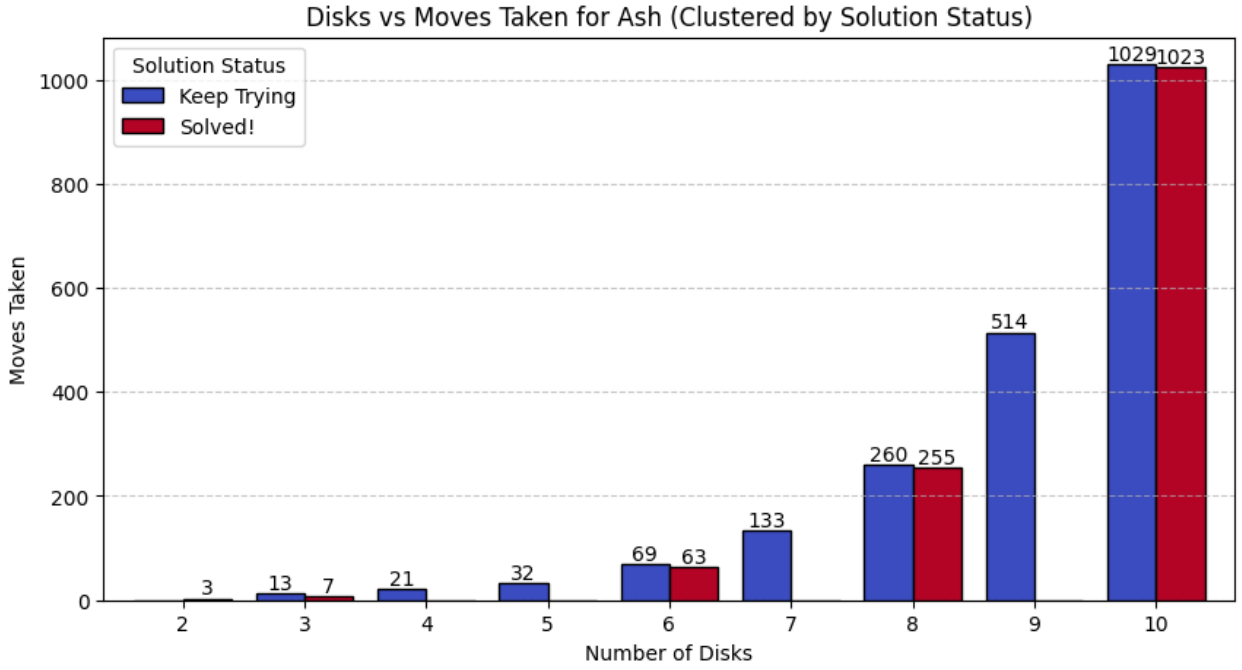

FIGURE 4.10: Disks vs Moves Taken For Ash

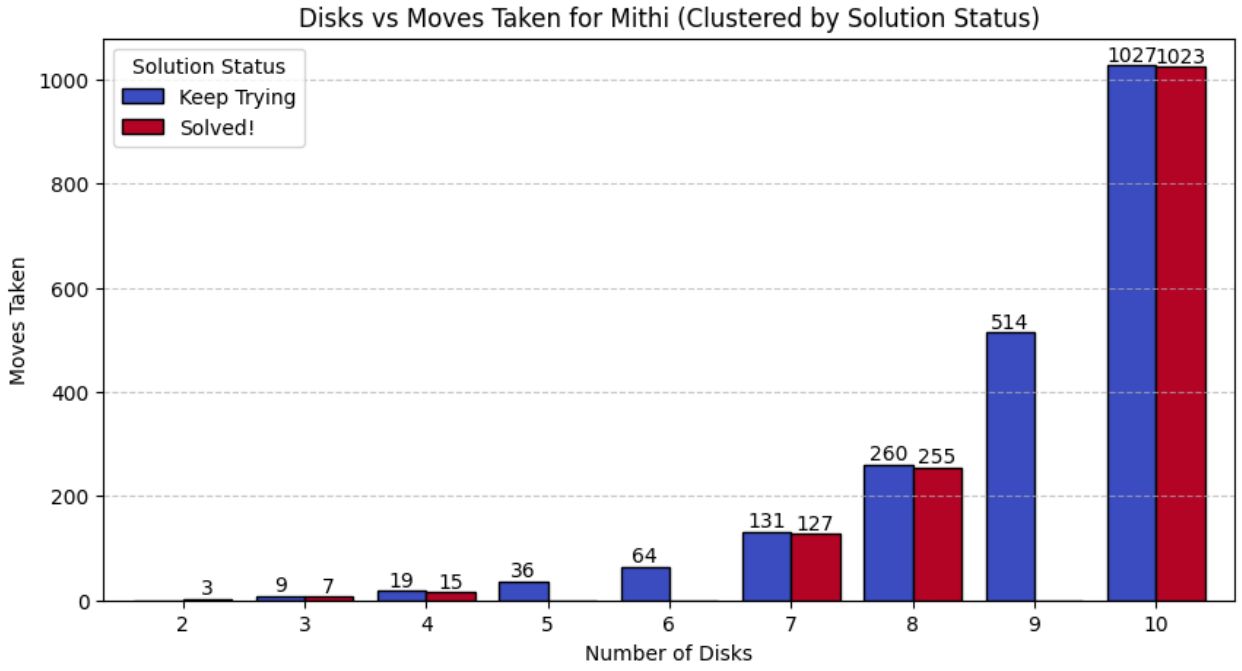

FIGURE 4.11: Disks vs Moves Taken For Mithi

4.3 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.

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

TABLE 4.4: Classification Report Summary for Josephus Problem using Random Forest Regression

FIGURE 4.12: Total People vs Wrong Moves For Max In Josephus Problem

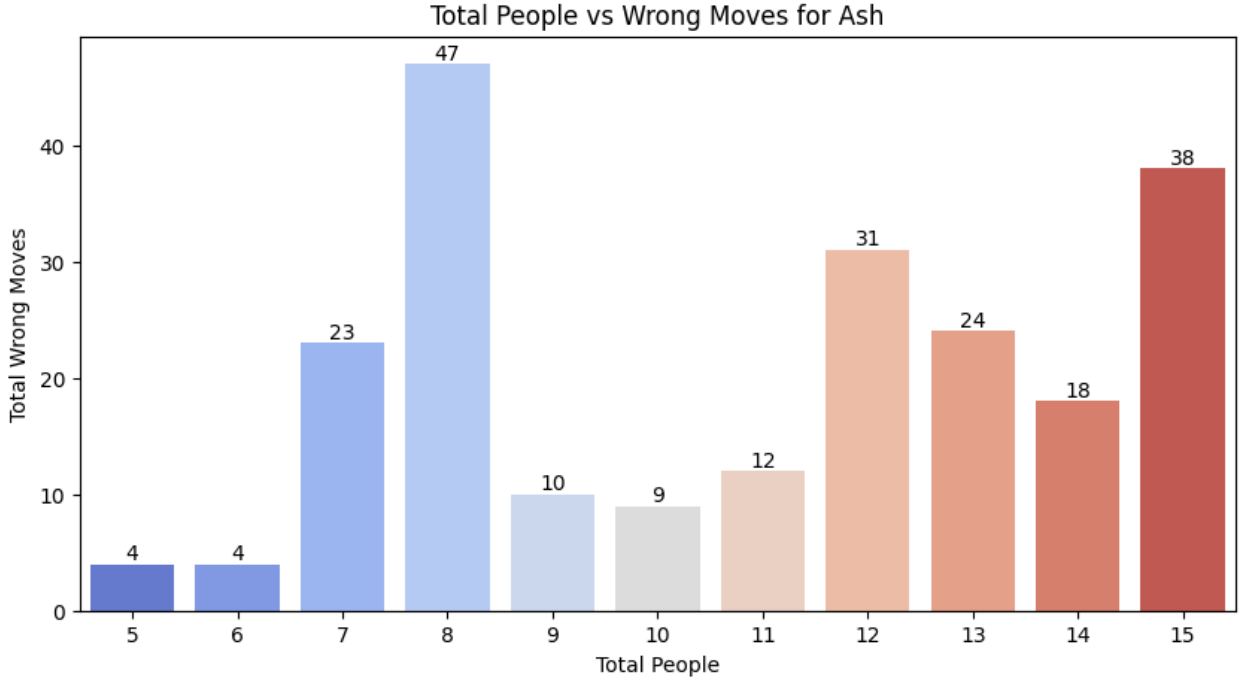

FIGURE 4.13: Total People vs Wrong Moves For Ash In Josephus Problem

FIGURE 4.14: Total People vs Wrong Moves For Mithi In Josephus Problem

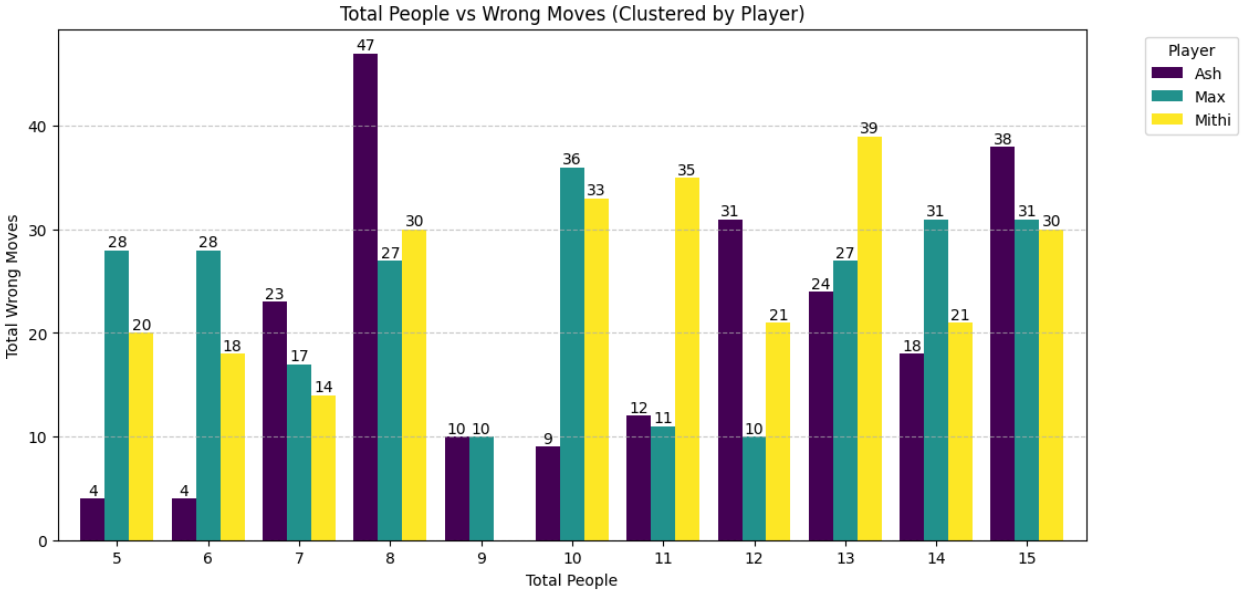

FIGURE 4.15: Total People vs Wrong Moves For Every Player(Clustered) In Josephus Problem

4.4 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.

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

TABLE 4.5: Classification Report for N-Queens Problem using Decision Tree Classifier

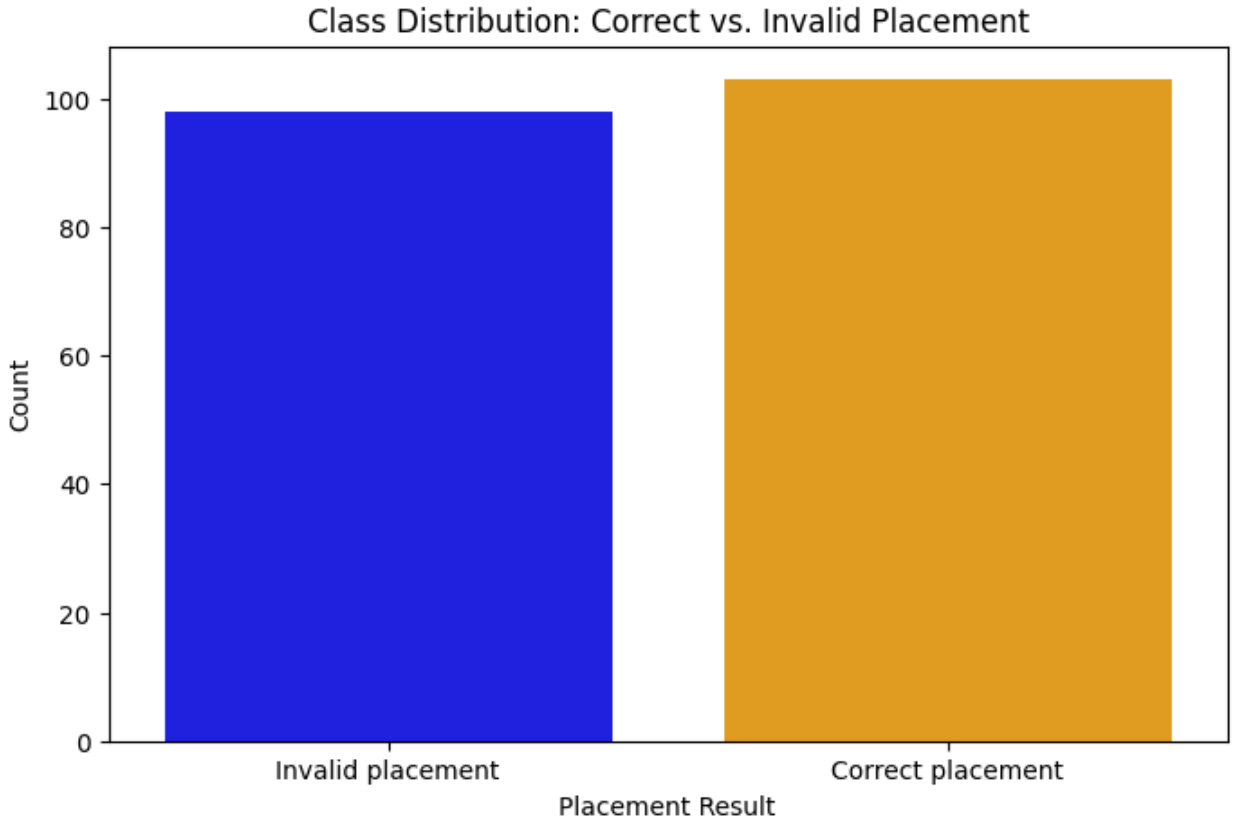

FIGURE 4.16: Correct vs Invalid Placement In N-Queens

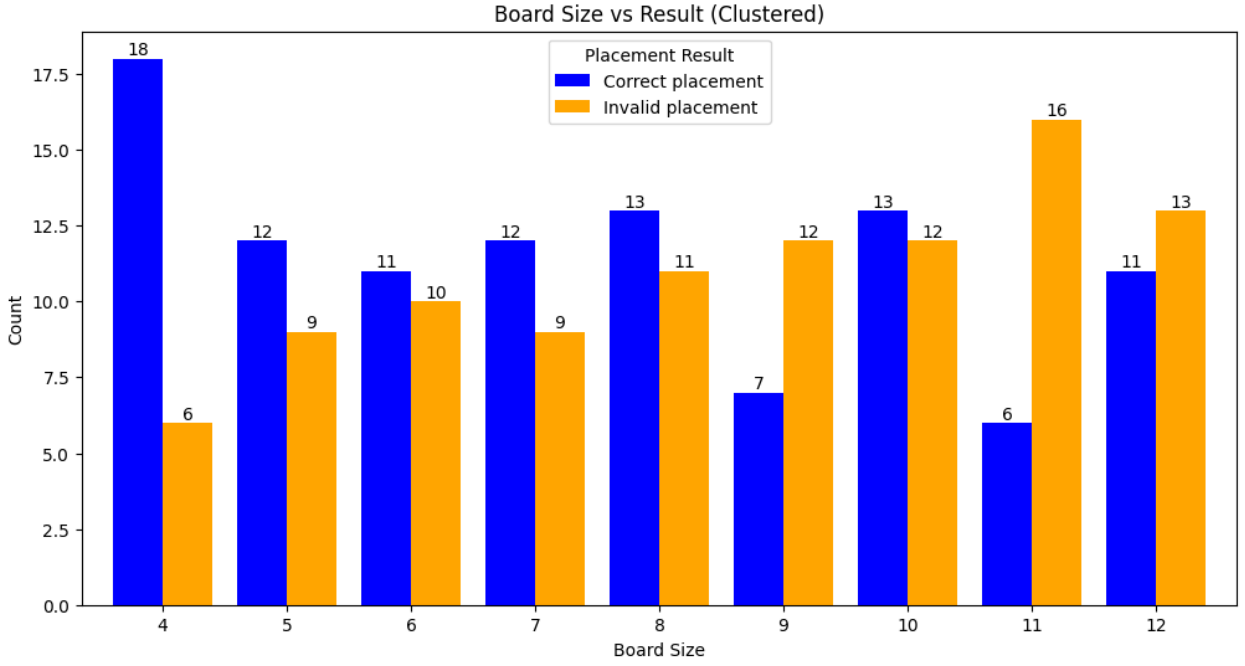

FIGURE 4.17: Board Size vs Result In N-Queens

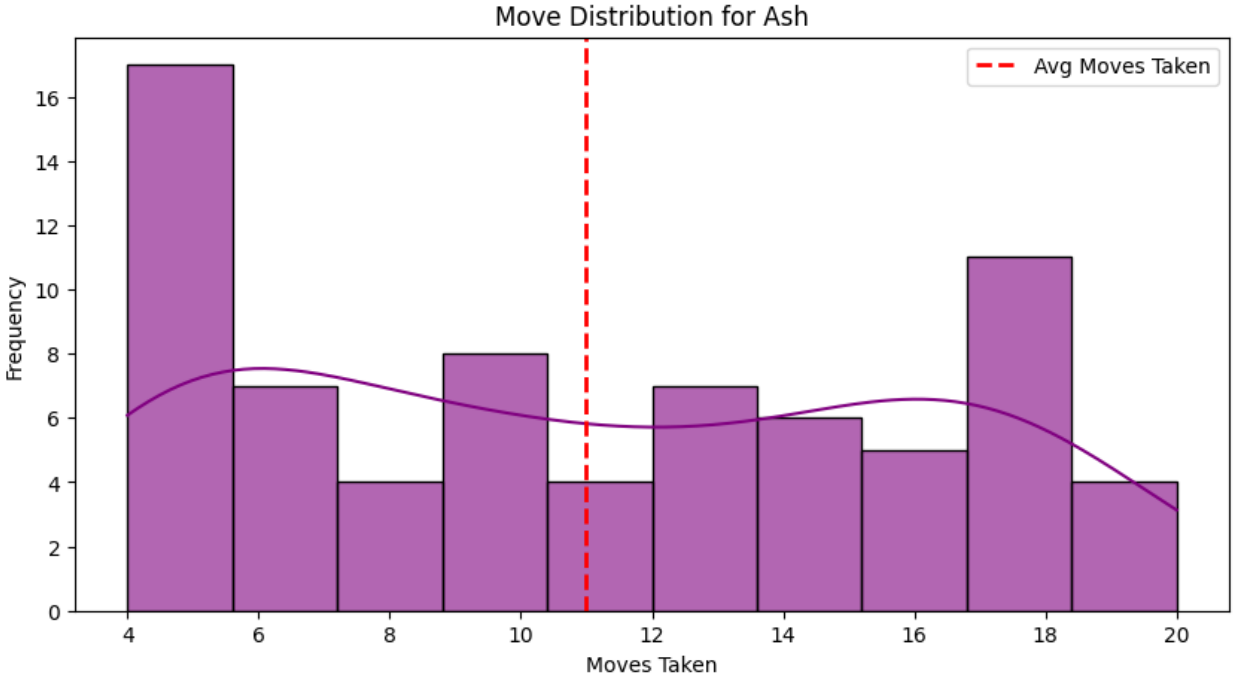

FIGURE 4.18: Move Distribution For Ash In N-Queens

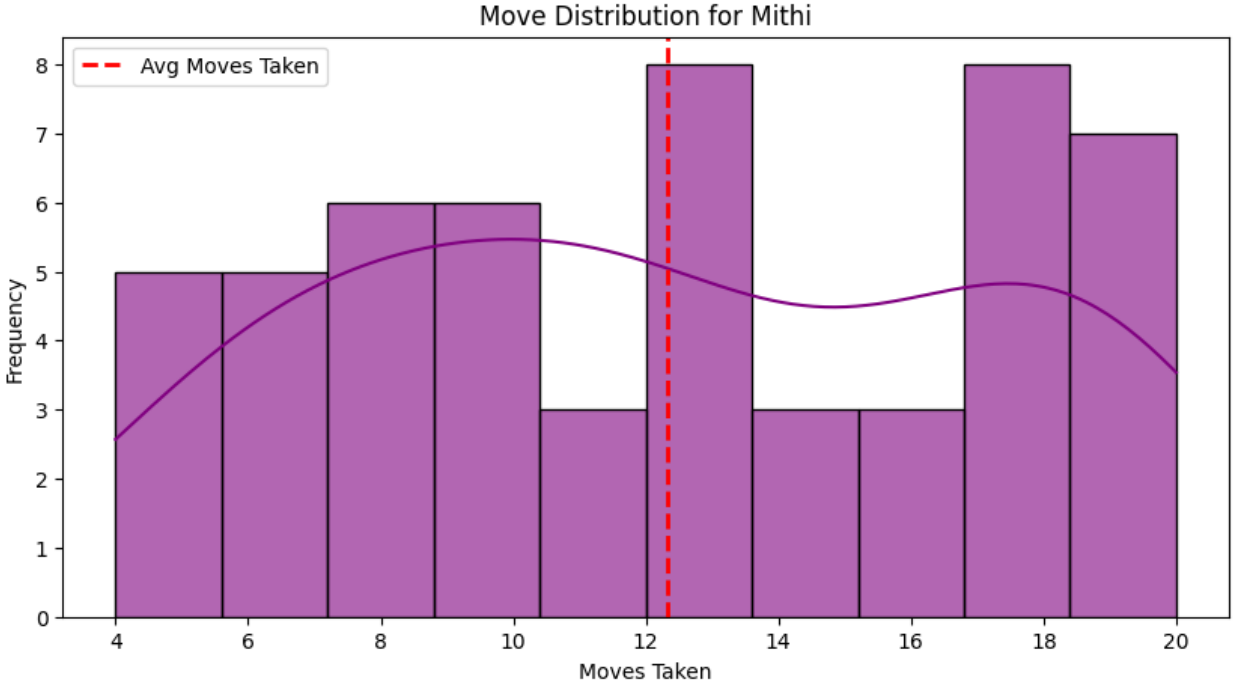

FIGURE 4.19: Move Distribution For Mithi In N-Queens

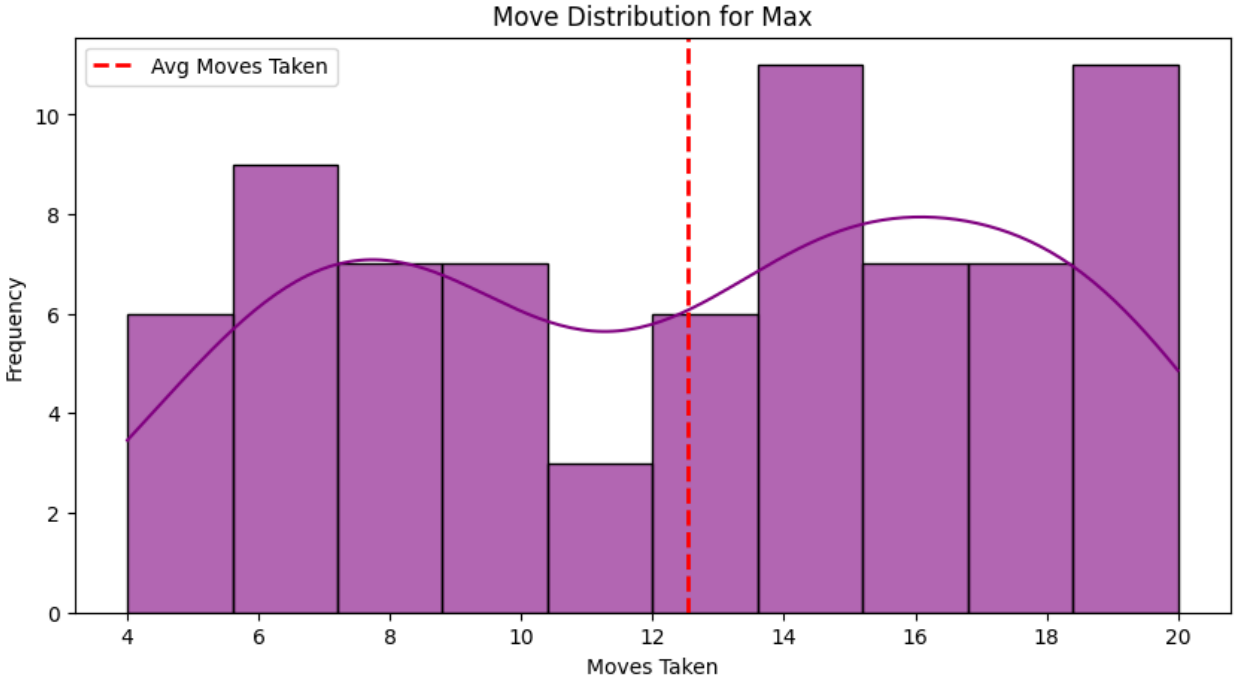

FIGURE 4.20: Move Distribution For Max In N-Queens

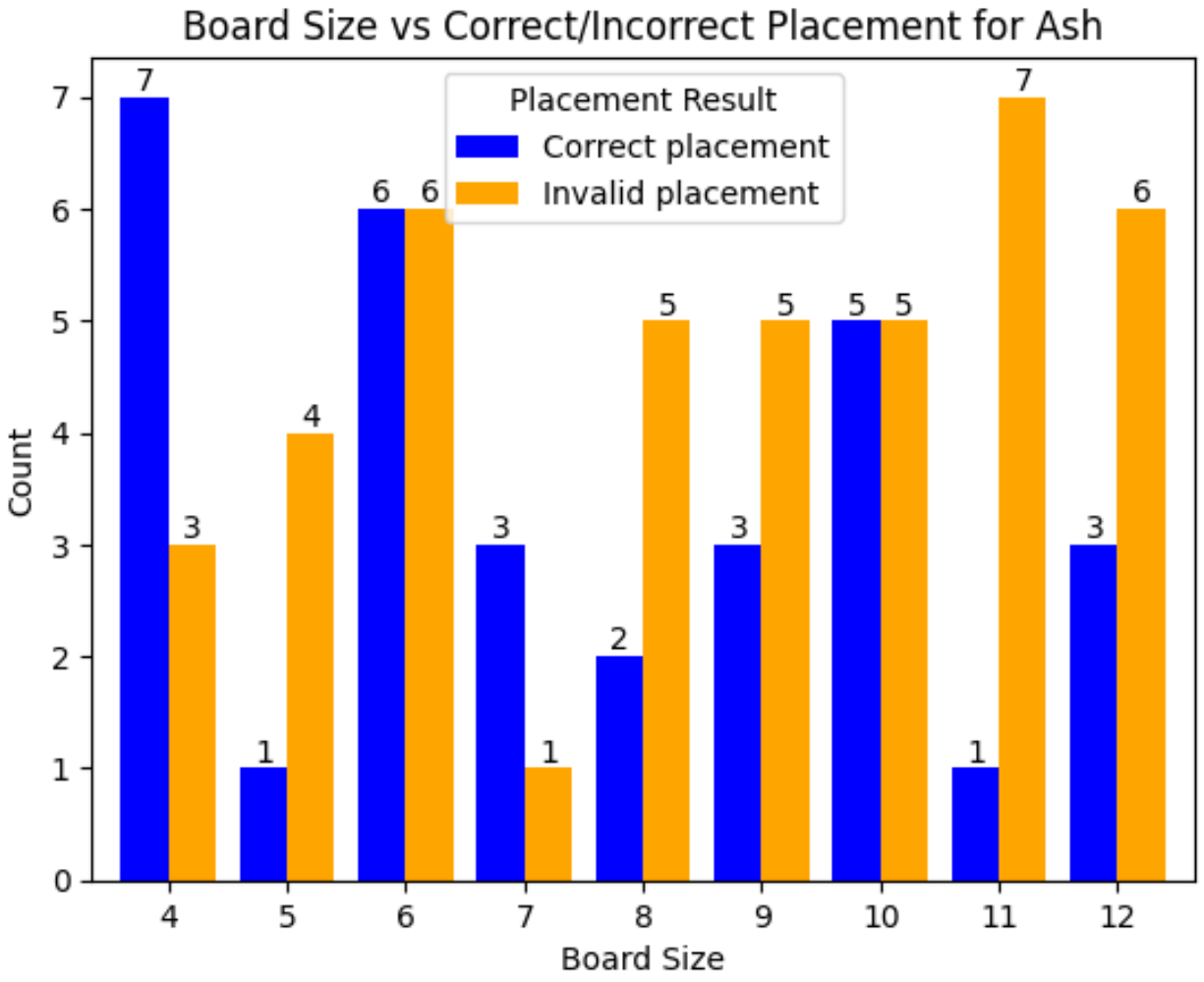

FIGURE 4.21: Board Size vs Result For Ash In N-Queens

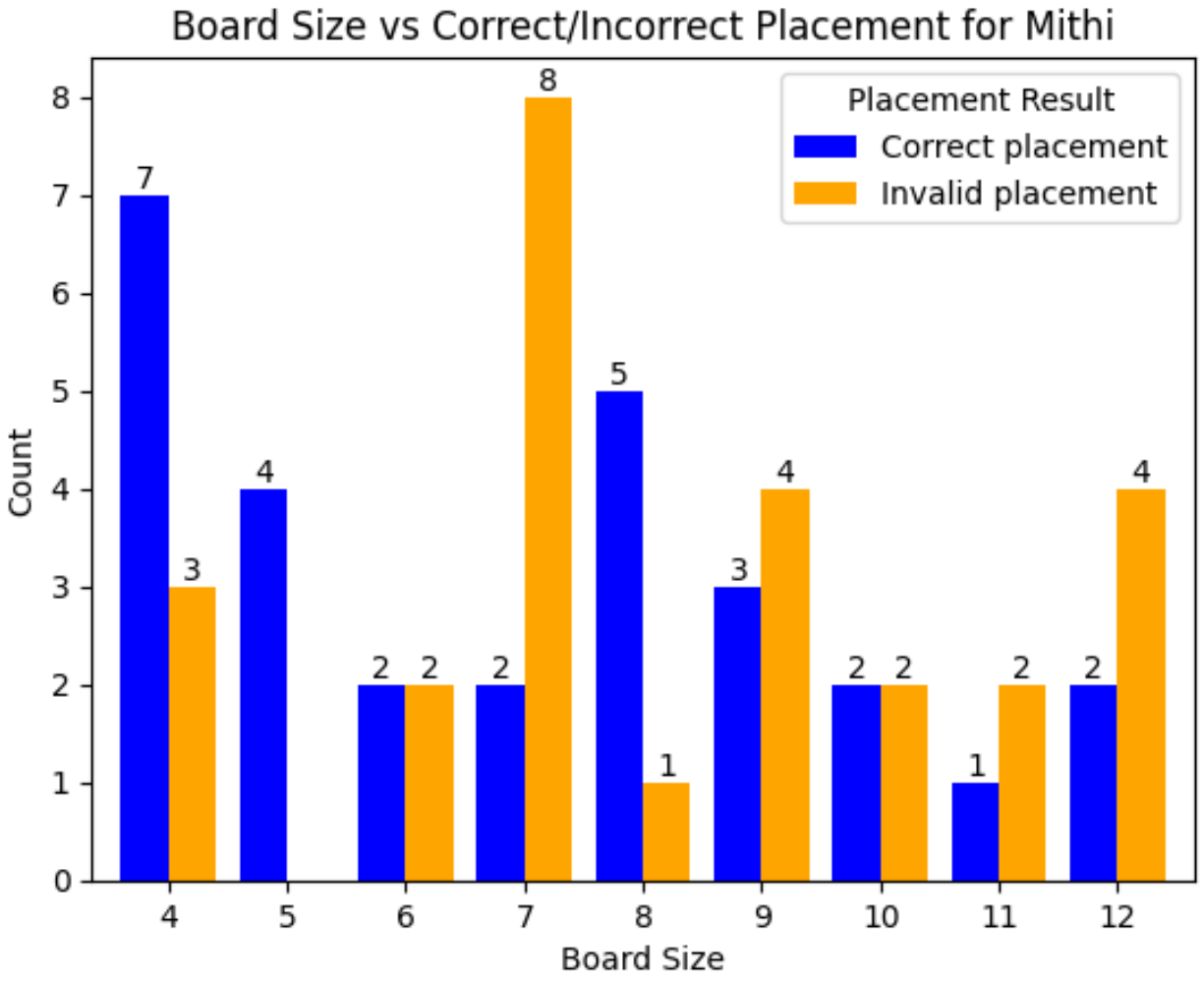

FIGURE 4.22: Board Size vs Result For Mithi In N-Queens

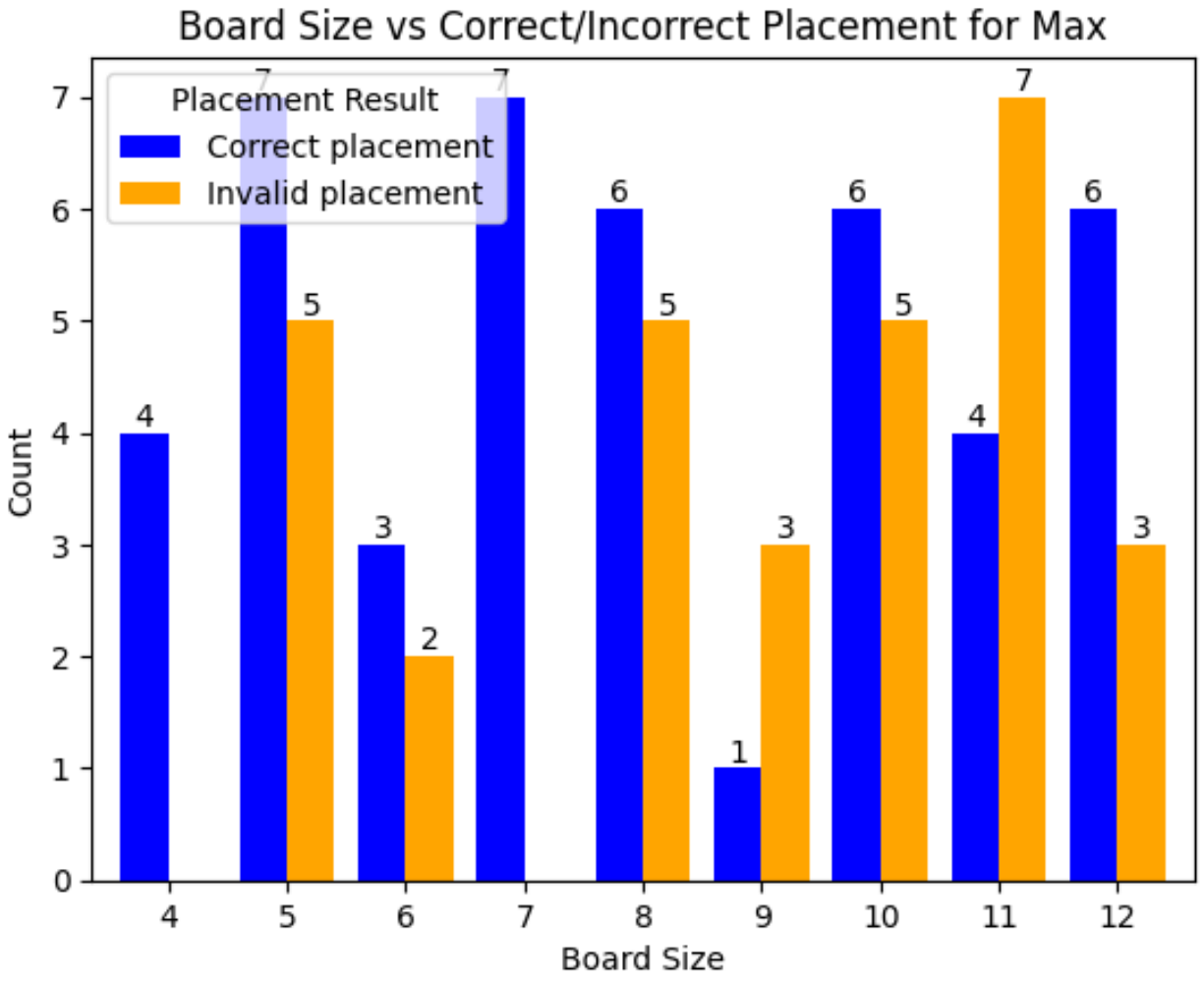

FIGURE 4.23: Board Size vs Result For Max In N-Queens

CHAPTER 5

IMPACTS PERTAINING TO SOCIETY, HEALTH, SAFETY, LEGAL, ENVIRONMENT AND CULTURE

The Animation Game Engine for Interactive Presentation of Educational Media holds immense potential to create positive, transformative impacts across various dimensions, including societal, health, safety, legal, environmental, and cultural aspects. By revolutionizing traditional learning methods, this innovative tool enhances accessibility, inclusivity, and engagement in education, ensuring that learners of all backgrounds have equal opportunities to grasp complex concepts.

By replacing lecture-based, passive learning methods with interactive animations and real-time feedback, the engine makes learning more engaging, immersive, and effective. Students who struggle with abstract ideas or text-heavy explanations can visualize concepts dynamically, reinforcing comprehension and retention. This approach bridges the gap between theory and application, ensuring that learners not only memorize information but also understand its practical significance. The ability to personalize learning paths further enhances the educational experience, catering to individual student needs and learning styles.

5.1 SOCIETAL IMPACTS

From a societal perspective, the animation engine is a powerful tool for democratizing education by making advanced learning technologies accessible to students from diverse socioeconomic backgrounds. In many underserved regions, access to high-quality educational tools is limited due to financial constraints, lack of infrastructure, or outdated teaching methods. Traditional e-learning platforms often require expensive hardware or high-speed internet, making them inaccessible to students in rural or low-income communities.

This animation engine addresses these challenges by offering an efficient, lightweight solution that can run smoothly on basic hardware without sacrificing quality. Schools in developing nations, remote villages, or economically disadvantaged areas can integrate interactive learning without the need for expensive technological upgrades. By leveling the playing field, this tool promotes educational equity, ensuring that all students, regardless of background, receive the same high-quality learning experience.

Additionally, the engine's real-time feedback system enables adaptive learning, allowing educators to identify and address individual students' weaknesses in a personalized and efficient manner. This feature is particularly beneficial for students who require additional support, ensuring that no learner is left behind. Over time, this approach fosters a more skilled, knowledgeable, and confident generation, contributing to social mobility, economic growth, and overall societal progress.

5.2 HEALTH IMPACTS

The animation engine enhances student engagement, encouraging learners to participate actively rather than passively consuming information. Traditional learning methods often lead to cognitive fatigue and disengagement, making it difficult for students to retain information. This engine mitigates these issues by offering an interactive and visually stimulating learning experience, keeping students mentally engaged and focused.

Cognitive research suggests that visual and interactive learning significantly improve memory retention, problem-solving abilities, and critical thinking skills. By incorporating animations, simulations, and interactive exercises, the engine stimulates multiple areas of the brain, reinforcing learning through visual, auditory, and kinesthetic engagement. This multisensory approach is particularly beneficial for students with learning disabilities or attention-related challenges, helping them grasp difficult concepts more effectively.

Additionally, the real-time feedback system provides students with instant encouragement and support, reducing frustration and anxiety associated with learning difficult subjects. Personalized feedback allows students to progress at their own pace, promoting a stress-free learning environment that prioritizes mental well-being. This approach helps foster self-confidence and motivation, leading to improved academic performance and long-term success.

5.3 SAFETY AND LEGAL IMPACTS

From a safety and legal standpoint, the animation engine is designed with strict security and data privacy measures to protect student and educator information. In an age where data breaches and cybersecurity threats are increasingly common, ensuring secure learning environments is paramount.

The engine complies with global data protection regulations, ensuring that user data remains confidential and secure. Students' interactions, quiz responses, and learning progress are stored in a safe, encrypted format, preventing unauthorized access or misuse. By adhering to industry-leading cybersecurity protocols, the system builds trust among educators, students, and institutions, promoting a safe digital learning ecosystem.

Additionally, the engine strictly adheres to intellectual property laws, ensuring that all third-party content, open-source assets, and educational materials are used ethically and responsibly. This commitment to legal compliance fosters fair use practices and protects the rights of content creators, further strengthening the integrity of the platform.

5.4 ENVIRONMENTAL IMPACTS

The animation engine aligns with global sustainability goals by offering a resource-efficient, low-energy solution for digital education. Unlike high-end learning platforms that require powerful computing resources and infrastructure, this engine is optimized to run smoothly on basic hardware, thereby reducing the need for frequent technological upgrades and significantly minimizing electronic waste (e-waste). This directly supports a reduction in hardware obsolescence and contributes to a sustainable digital environment.

Furthermore, the engine actively promotes eco-friendly educational practices by reducing reliance on paper-based learning materials. With the digital delivery of complex animations, interactive quizzes, and algorithm visualizations, the need for physical textbooks, handouts, and printed materials is significantly lowered. This reduction in paper usage minimizes paper waste, lowers printing and distribution costs, and helps reduce the carbon footprint associated with traditional educational systems.

Additionally, as the engine supports scalable and digital content delivery, it eliminates the logistical challenges of transporting physical educational materials, further cutting down emissions related to logistics and transportation. By integrating sustainability principles into its design, the animation engine not only fosters a greener, more environmentally responsible learning environment but also contributes to long-term ecological balance. This aligns well with modern educational institutions' increasing focus on environmental stewardship and responsible technology adoption.

5.5 CULTURAL IMPACTS

Education should be inclusive and culturally relevant, ensuring that students from different backgrounds, languages, and traditions feel represented in the learning process. The animation engine is highly customizable, allowing educators to adapt content to suit the linguistic, cultural, and contextual needs of diverse student populations.

For example, educators can incorporate localized themes, historical references, and culturally relevant examples into the animations, making lessons more relatable and engaging for students. This adaptability ensures that learning remains relevant across various educational systems, geographic regions, and cultural traditions.

Additionally, the engine's accessibility features, such as multilingual support, text-to-speech functionality, and customizable interfaces, cater to students with different linguistic and physical abilities. By prioritizing inclusive design, the engine fosters a sense of belonging and equal opportunity for all learners, regardless of their cultural, linguistic, or physical backgrounds.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

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. Andhika Kusheryanto, Anis Mirza, Ari Syaripudin & Deanna Durbin Hutagalung. (2024) Development of the Story of Life: A Narrative and Educational Game Using the Godot Engine for Android.

2. Andreas Kerren. (2012) Visualizations and Animations in Learning Systems.

3. Ariel Manzur & George Marques. (2018) Godot Engine Game Development in 24 Hours, Sams Teach Yourself: The Official Guide to Godot 3.0.

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

5. Carneiro, D., and Carvalho, M. (2022) 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.

6. Cakiroglu, Y., and Ergun, M. (2016) The Role of Interactive Animations in Enhancing Problem-Solving Skills.

7. Genady Kogan, Hadas Chassidim & Irina Rabaev. (2024) The efficacy of animation and visualization in teaching data structures: a case study.

8. Henry W. Robbins, Samuel C. Gutekunst, David B. Shmoys & David P. Williamson. (2023) GILP: An Interactive Tool for Visualizing the Simplex Algorithm.

9. Jeetha, K.P.K. (2021) Animation for Learning: Enhancement of Learning through Animation: A Review of Literature. International Research Journal of Modernization in Engineering Technology and Science, Volume: 03/Issue: 01, January.

10. John T. Stasko. (1997) Using student-built algorithm animations as learning aids.

11. Ken Perlin, Zhenyi He & Karl Rosenberg. (2018) Chalktalk : A Visualization and Communication Language – As a Tool in the Domain of Computer Science Education.

12. LADISLAV VÉGH & Veronika Stoffová. (2017) Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings.

13. Maithili Dhule. (2022) Beginning Game Development with Godot.

14. Mahesh Ranaweera Qusay & H. Mahmoud. (2024) Deep Reinforcement Learning with Godot Game Engine.

15. Martin Krajčovič, Gabriela Gabajová, Beáta Furmannová, Vladimír Vavrík, Martin GašoORCID & Marián Matys. (2021) A Case Study of Educational Games in Virtual Reality as a Teaching Method of Lean Management.

16. Maximiliano Paredes-Velasco, J. Ángel Velázquez-Iturbide & Mónica Gómez-Ríos. (2022) Augmented reality with algorithm animation and their effect on students' emotions.

17. Sarthak Goel, Vanshika Varshney, Shubham Dikshant, Akhil Sharma & Sherish Johri. (2023) A Review of The Algorithm Visualization Field.

18. Shute, V.J., and Ke, F. (2012) Games, Learning, and Assessment: Progressive Design of Games to Facilitate Learning and Assessment. In Assessment in Game-Based Learning, Springer, pp.43-58.

19. Steven Hansen & N. Hari Narayanan. (2000) On the Role of Animated Analogies in Algorithm Visualizations.

20. Steven Hansen, N. Hari Narayanan & Mary Hegarty. (2002) Designing Educationally Effective Algorithm Visualizations.

21. Su, S., Zhang, E., Denny, P., and Giacaman, N. (2021) 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.

22. Wilson, A., and Martin, L. (2019) Applications of Interactive Media in Education: Study on Adaptive Learning.

23. Wu, B., and Richards, B. (2021) An Overview of Gamification in E-Learning and Its Effects on Motivation and Learning. Computers in Human Behavior Reports, 3, p.100086.