International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 09 September-2026 | Impact Factor: 3.5

Elemental Strategy: Intelligent Board Game

Mayank R. Bhagat[1], Sujal B. Kaithal[2], Tejas Y. Rakhade[3], Vaibhav S. Khapekar[4], Prof. Rakesh M. Moharle[5], Dr. Sushama V. Telrandhe[6]

[1234]U.G. Students, Department of Computer Science and Engineering, Gurunanak Institute of Engineering and Technology, Nagpur, India.
[5]Assistant professor, Department of Computer Science and Engineering, Gurunanak Institute of Engineering and Technology, Nagpur, India.
[6]Associate professor, Department of Computer Science and Engineering, Gurunanak Institute of Engineering and Technology, Nagpur, India.

How to Cite this Article:
Bhagat, M. R., Kaithal, S. B., Rakhade, T. Y., Khapekar, V. S., Moharle, R. M. & Telrandhe, S. V. (2026). Elemental Strategy: Intelligent Board Game. International Journal of Creative and Open Research in Engineering and Management, 02(9), 1-9. https://doi.org/10.55041/ijcope.v2i9.021

https://doi.org/10.55041/ijcope.v2i9.021

Abstract

Video games have evolved from simple entertainment applications into interactive systems that can improve problem-solving, decision-making, and strategic thinking. Artificial Intelligence (AI) plays an important role in modern games through techniques such as pathfinding, decision-making, finite state machines, and dynamic difficulty adjustment. This project presents a 2D grid-based escape game in which the player controls a character that must move from a starting position to an escape door while overcoming different obstacles such as ice, fire, water, and walls. The player is required to make strategic decisions and use appropriate elements, such as fire to melt ice and water to overcome fire-based obstacles, to progress through the game. Pathfinding techniques such as A* and Dijkstra can be used to determine efficient routes while considering obstacles and game conditions. Finite State Machines can also be applied to control intelligent game objects or enemies, while Dynamic Difficulty Adjustment can be used to increase the challenge according to the player's performance. Sound effects provide additional feedback and improve player interaction and engagement. The proposed game combines grid-based movement, obstacle solving, elemental interactions, pathfinding, decision-making, and audio feedback to create an interactive and challenging 2D escape game.

Introduction

Computer games have evolved significantly from simple forms of entertainment into interactive applications that can support problem-solving, decision-making, learning, and strategic thinking. The development of Artificial Intelligence (AI) has played an important role in this evolution by enabling games to provide intelligent, responsive, and adaptive experiences. AI techniques such as pathfinding, decision-making, search algorithms, Finite State Machines (FSM), and Dynamic Difficulty Adjustment (DDA) are commonly used to improve gameplay and player engagement.

Game engines provide the necessary tools and components for developing games, including graphics, physics, audio, rendering, and AI systems. In 2D games, AI can be used to control characters, determine suitable movements, respond to game events, and manage interactions with the game environment. Among various AI techniques, pathfinding is particularly important because it helps a character or non-player character find an efficient route from a starting point to a target while avoiding obstacles. Algorithms such as A* and Dijkstra's algorithm are widely used for this purpose.

Player experience is another important aspect of game development. A game should provide an

appropriate level of challenge so that players remain interested without becoming bored or frustrated. Dynamic Difficulty Adjustment can be used to modify the difficulty according to the player's performance. Similarly, decision-making techniques allow players or game agents to select appropriate actions based on the current game state.

Based on these concepts, this project proposes a 2D grid-based escape game in which the player controls a character that starts at a fixed position and must reach an escape door. The game board consists of small cells through which the character can move. Different obstacles, including ice, fire, water, and walls, are placed along the path. The player must use appropriate elements, such as fire to melt ice and water to overcome fire-related obstacles, while selecting a suitable path toward the escape door. Sound effects are also incorporated to provide feedback and make the gameplay more interactive and engaging.

The proposed game combines concepts of grid-based movement, obstacle avoidance, elemental interaction, pathfinding, decision-making, and audio feedback to create a challenging and interactive 2D escape experience. The project demonstrates how AI and game-development techniques can be combined to improve gameplay while providing players with opportunities to develop logical thinking and problem-solving skills.

Project Objective

The main objective of this project is to design and develop an interactive 2D escape board game using Java, in which the player controls a character and guides it from a starting position to an escape door. The game uses a grid-based board made up of small boxes for character movement. Various obstacles such as ice, fire, water, and walls are placed along the path, requiring the player to think strategically and use suitable elements to overcome them. Elements such as fire and water are provided to interact with and overcome specific obstacles. The project aims to provide an engaging and challenging gameplay experience by combining movement, obstacle solving, strategic decision-making, and problem-solving. Additionally, sound effects are incorporated to make the game more interactive and enhance the overall gaming experience. The complete game is developed using Java, applying concepts such as object-oriented programming, graphics, event handling, and game logic.

Literature Survey

The reviewed research papers show that Artificial Intelligence and game algorithms can be effectively used to improve gameplay, decision-making, movement, and player interaction. Different techniques such as A* path-finding, Minimax with Alpha-Beta pruning, Monte Carlo Tree Search, Artificial Neural Networks, decision trees, and state machines have been applied to different types of games. These techniques help game characters find suitable paths, make decisions, respond to game situations, and provide challenging gameplay. The studies also demonstrate that algorithm-based game design can make games more interactive and can provide different levels of difficulty and strategic challenges.

Based on these studies, the proposed project focuses on developing a Java-based grid escape game in which the player moves a character from a starting point to an escape door while overcoming obstacles such as ice, fire, water, and walls. The player must use appropriate elements, such as fire or water, to overcome specific obstacles and select a suitable path to reach the destination. The existing research provides useful concepts for grid movement, path-finding, decision-making, and game logic,

while the proposed project combines these ideas with element-based obstacle interaction and sound effects to create a simple, interactive, and challenging escape game. Thus, the literature provides a foundation for the development of the proposed game while also showing an opportunity to create a different gameplay approach centred on obstacle solving and escape-based objectives.

Methodology

The proposed project uses a Java-based 2D grid game methodology, inspired by the game-development and AI techniques discussed in the reviewed research papers. The game is designed around a grid-based board, where the playing area is divided into small boxes or cells. The player character starts from a predefined starting position and must reach the escape door by selecting a suitable path. Different obstacles, including ice, fire, water, and walls, are placed on the board to make the path challenging. The movement and board can be represented using nodes or cells, similar to the grid representation used in path-finding approaches. The research papers discuss algorithms such as A*, Dijkstra, Depth-First Search, and Breadth-First Search for finding paths in game environments; these concepts provide a basis for handling movement and path selection in the proposed game.

The game also uses an obstacle-interaction methodology, where the player needs to use appropriate elements to overcome obstacles. For example, fire can be used to melt ice, while water can be used to overcome fire. Therefore, the player must not only find a path but also make appropriate decisions about which element to use. This relates to the decision-making concept discussed in the reviewed research, where game characters process information about their environment and select an appropriate action. Decision trees and state machines are examples of techniques described for game decision-making.

The game follows a continuous game-loop methodology, in which the game initializes the board and required resources, accepts player input, updates the character's position and game state, checks obstacle and element interactions, and continuously renders the updated game scene. The reviewed game-engine research describes a similar process in which a scene is initialized, resources are loaded, and the game loop continuously updates and renders the active scene. In the proposed Java implementation, this methodology can be adapted to manage the board, character, obstacles, elements, escape door, and sound effects.

Finally, the methodology focuses on providing an interactive and challenging gameplay experience. The player must analyze the board, select an appropriate route, overcome obstacles using available elements, and finally reach the escape door. Sound effects are incorporated for important game events to improve interaction and feedback. The overall methodology therefore combines grid-based movement, path selection, decision-making, obstacle interaction, game-state management, graphics rendering, and sound effects to develop the proposed Java-based escape game. The reviewed research supports the use of path-finding and decision-making concepts, while the specific combination of element-based obstacle solving and escape gameplay is the proposed project's own implementation.

Tools and Platforms to be Used

The proposed game will be developed using Java as the main programming language. Java Swing and Java AWT will be used to create the graphical user interface and the grid-based game board, including the character, obstacles, elements, and escape door. Java 2D Graphics

will be used for drawing and displaying game objects and their movements. The Java Sound API will be used to implement sound effects for activities such as character movement, obstacle interaction, collecting elements, and reaching the escape door. The Java Development Kit (JDK) will be required to compile and execute the program. An Integrated Development Environment (IDE) such as IntelliJ IDEA, Eclipse, or NetBeans can be used for writing, debugging, and testing the Java code. Image files such as PNG or JPG can be used for game characters and objects. The game can be developed and executed on platforms such as Windows or Linux, making the overall system simple and suitable for a college-level Java game project.

Advantages

The proposed Java-based escape game has several advantages. It provides an interactive and entertaining gaming experience by combining character movement, obstacles, elements, and an escape objective. The grid-based board makes movement simple and easy to understand. The different obstacles such as ice, fire, water, and walls encourage the player to think strategically and select appropriate paths. The use of elements such as fire to melt ice and water to overcome fire adds a problem-solving aspect to the game. Sound effects improve player interaction and provide feedback during important game events. Since the game is developed using Java, it demonstrates practical concepts such as object-oriented programming, graphics, event handling, and game logic. The project can also be extended with additional levels, obstacles, characters, and intelligent path-finding algorithms.

Disadvantages

The project also has some limitations. Since it is a college-level 2D game, the graphics and gameplay may be simpler compared with commercial games. The game currently depends mainly on predefined obstacles and rules, which may reduce variety and replayability. Developing advanced AI-based movement or adaptive difficulty can increase the complexity of the Java program. The game may also require additional image and sound resources to provide a more attractive experience. If the board size and number of obstacles become very large, path-finding and game processing may require more computational resources. Additionally, the game is primarily designed for a desktop environment, so mobile or online multiplayer support would require additional development.

Result

The proposed Java-based escape game is expected to provide an interactive, challenging, and problem-solving-oriented gaming experience. Based on the reviewed research, game development can be improved through suitable game logic, path-finding, decision-making, and adaptive techniques. The proposed game applies these ideas in a simpler form by providing a grid-based board where the player moves from a starting point toward an escape door while overcoming ice, fire, water, and wall obstacles. The use of appropriate elements to overcome obstacles requires the player to make decisions and select suitable paths. Similar to the referenced studies, the project focuses on improving player engagement and creating an enjoyable gameplay experience. The implementation of graphics, game logic, character movement, obstacle interaction, and sound effects is expected to make the game more interactive. The referenced research also shows that intelligent game systems can adapt and improve their performance over time, providing

a basis for further enhancement of the proposed game.

Future Scope

The proposed game has significant scope for future development. Artificial Intelligence and path-finding algorithms such as A* can be incorporated to automatically determine suitable paths through the grid. The game can also be enhanced with adaptive difficulty, where the difficulty level changes according to the player's performance, an approach discussed in the referenced research on Dynamic Difficulty Adjustment. More advanced AI techniques, such as Dynamic Scripting, Behaviour Trees, and Finite-State Machines, can be integrated to create more intelligent game behaviour. The number of levels, obstacles, characters, and usable elements can also be increased to improve replayability. Future versions can include player performance analysis, automatic level generation, improved graphics, background music, multiplayer functionality, and mobile-platform support. The referenced research specifically identifies integration of Behaviour Trees and Finite-State Machines with adaptive systems and the development of easier tools for implementing dynamic difficulty as promising future research areas.

Conclusion

The proposed project successfully applies the concepts identified from the reviewed research on AI, path-finding, decision-making, and game development to create a simple and interactive Java-based 2D escape game. The game provides a grid-based environment in which the player must move from the starting point to the escape door while overcoming obstacles such as ice, fire, water, and walls. The use of different elements to overcome specific obstacles introduces planning and decision-making into the gameplay. The reviewed studies show that techniques such as A*, Dijkstra, Minimax, Monte Carlo Tree Search, Artificial Neural Networks, and Finite-State Machines can improve game intelligence, movement, and player interaction.

In the proposed project, these research concepts provide a foundation for developing effective game logic, movement, obstacle handling, and decision-making. The addition of sound effects and interactive elements further improves the player's gaming experience. Similar to the referenced research, the project demonstrates that even a relatively simple 2D game can provide a challenging experience when appropriate planning and game mechanics are introduced. Future integration of A* path-finding, Finite-State Machines, or Dynamic Difficulty Adjustment could make the game more intelligent and adaptive. The reviewed research also indicates that adaptive techniques can improve player engagement when difficulty is appropriately adjusted to the player's performance.

Overall, the project demonstrates the practical application of Java programming, 2D game development, problem-solving, path selection, and interactive game mechanics. It provides a foundation that can be further extended with more advanced AI techniques, additional levels, improved graphics, and adaptive gameplay.

References

1. Graham, R., McCabe, H., & Sheridan, S. (2003). Pathfinding in Computer Games. *ITB Journal*, Vol. 4, Issue 2.

2. Cui, X., & Shi, H. (2011). *A-Based Pathfinding in Modern Computer Games.* International Journal of Computer Science and Network Security*, 11(1).

3. Permana, S. D. H., Bintoro, K. B. Y., Arifitama, B., & Syahputra, A. (2018). *Comparative Analysis of Pathfinding Algorithms A, Dijkstra, and BFS on Maze Runner Game.* International Journal of Information System & Technology*.

4. Buckland, M. (2005). Programming Game AI by Example. Wordware Publishing.

5. Millington, I. (2019). AI for Games. 3rd ed., CRC Press.

6. Gregory, J. (2018). Game Engine Architecture. AK Peters/CRC Press.

7. Ranjitha, M., Nathan, K., & Joseph, L. (2020). Artificial Intelligence Algorithms and Techniques in the Computation of Player-Adaptive Games. *Journal of Physics: Conference Series*, 1427(1), 012006.