International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 11 November-2026 | Impact Factor: 3.5



Mushroom Survival Game: A Unity-Based 3D Survival Game with Health Management and Time-Based Objectives

Basavaprabhu
Department of CSE
Akshaya Institute of Technology
Tumakuru, India

A. Punith Kumar
Department of CSE
Akshaya Institute of Technology
Tumakuru, India

L. Lohith
Department of CSE
Akshaya Institute of Technology
Tumakuru, India

B. Kiran Patil
Department of CSE
Akshaya Institute of Technology
Tumakuru, India

Ms. Thippamma S
Associate Professor, Project Guide
Department of CSE
Akshaya Institute of Technology
Tumakuru, India

How to Cite this Article:
Basavaprabhu, , Kumar, A. P., Lohith, L. & Patil, B. K. (2026). Mushroom Survival Game: A Unity-Based 3D Survival Game with Health Management and Time-Based Objectives. International Journal of Creative and Open Research in Engineering and Management, 02(11), 1-9. https://doi.org/10.55041/ijcope.v2i11.164

License:
This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.
© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.





https://doi.org/10.55041/ijcope.v2i11.164

***Abstract*—This paper presents the development of the Mush-room Survival Game, a three-dimensional survival game created using Unity and C#. The game allows players to explore a forest environment, collect safe mushrooms, and avoid poisonous mushrooms. The system includes third-person player movement, mushroom collection tracking, health management, a countdown timer, and a day-night cycle. The player wins by collecting the required number of safe mushrooms before time runs out. The game ends if the player's health reaches zero or the timer expires. This project demonstrates the integration of object interaction, environmental hazards, and time-based objectives in an interactive 3D game environment.**

***Index Terms*—Unity, C#, 3D Game Development, Survival Game, Mushroom Collection, Health Management, Game Design.**

I. Introduction

Three-dimensional games provide interactive environments where players can explore virtual worlds and complete differ-ent objectives. Modern game engines provide tools for cre-ating environments, controlling characters, and implementing gameplay mechanics [1], [6].

The Mushroom Survival Game is developed using Unity and C#. The player explores a forest, collects safe mushrooms, and avoids poisonous mushrooms that reduce health. A countdown timer creates a time-based challenge, while a day-night cycle changes the environment's lighting.

The main goal is to collect the required number of safe mushrooms before time runs out while maintaining the player's health. The project combines character movement, object interaction, health management, and survival mechanics in a single game environment.

II. Problem Statement

A survival game requires clear objectives, meaningful challenges, and feedback to keep players engaged. Without suitable gameplay mechanics, the player's progress may be difficult to understand and the game may become less challenging.

This project addresses the development of a 3D forest survival game in which players must collect safe mushrooms, avoid poisonous mushrooms, manage their health, and complete the objective within a limited time. The system must provide clear information about collected mushrooms, remaining health, remaining time, and the final game outcome.

III. Objectives

The main objectives of this project are:

- To develop a 3D forest environment using Unity.
- To implement third-person player movement and camera control.
- To enable players to collect safe mushrooms.
- To introduce poisonous mushrooms as hazards.
- To implement a player health management system.
- To add a countdown timer and day-night cycle.
- To display victory when the collection objective is completed.
- To display Game Over when health reaches zero or the timer expires.

IV. Related Work

Game engines provide tools for rendering, animation, physics, and interaction. Unity supports three-dimensional game development and scripting through C# [1], [2].



Character controllers are commonly used to manage player movement, while colliders and trigger events support interaction between game objects [3]. User interface elements provide information such as health, timers, and objective progress.

Game design literature discusses the importance of player experience, gameplay mechanics, and interaction in creating engaging games [6]–[8]. These concepts provide background for designing the Mushroom Survival Game. Further literature review can compare this project with specific published survival games and game interaction studies.

V. Technologies Used

A. Unity Game Engine

Unity is used to create the forest environment, configure game objects, manage lighting, and implement gameplay features [1].

B. C# Programming Language

C# scripts control player movement, mushroom collection, health management, timer operation, and game-state transitions [2].

C. Character Controller

The Character Controller supports player movement and collision handling within the game environment.

D. Colliders and Triggers

Colliders and trigger events detect interactions between the player and mushrooms. These interactions support collection of safe mushrooms and detection of poisonous mushrooms [3].

E. User Interface

Unity UI and TextMeshPro display the mushroom counter, health information, timer, victory message, and Game Over screen [4].

VI. Proposed Methodology

The development process is divided into several stages.

A. Environment Creation

A forest environment is created in Unity using terrain, trees, vegetation, lighting, and mushroom objects. This environment provides the main area for exploration.

B. Player Movement

A third-person character is configured with a movement script and Character Controller. Camera control allows the player to view and navigate the environment.

C. Mushroom Collection

Safe mushrooms are configured with colliders and interaction logic. When the player approaches a collectible mushroom and presses the interaction key, the mushroom is collected and the counter is updated.

D. Poisonous Mushroom Detection

Poisonous mushrooms are configured as environmental hazards. Contact with these objects reduces the player's health through the health management system.

E. Health Management

The health system tracks the player's current health. The interface displays the current value. When health reaches zero, the game displays the Game Over screen.

F. Countdown Timer

The countdown timer tracks the remaining gameplay time. If the timer reaches zero before the collection objective is completed, the game enters the Game Over state.

G. Day-Night Cycle

A directional light changes its rotation and lighting properties over time to simulate changes in daylight and create variation in the forest environment.

H. Victory and Game Over

The game checks the player's collection progress and survival conditions. Collecting the required number of safe mushrooms results in victory. Reaching zero health or running out of time results in Game Over.

VII. System Design

The game uses a component-based structure in which individual scripts manage specific gameplay functions. The main components include the player controller, environment, mushroom interaction system, health system, timer, day-night controller, and user interface.

The overall gameplay flow is:

Player Exploration
↓
Mushroom Collection and Hazard Detection
↓
Health and Timer Updates
↓
Game-State Evaluation
↓
Victory or Game Over

The player explores the forest and interacts with mushrooms. The game updates the collection counter, health, and remaining time. Based on the objective and survival conditions, the system displays the appropriate final state.

VIII. Implementation

The game is implemented in Unity using C# scripts. Player movement is managed through a Character Controller, while the camera follows the player during exploration.

Safe mushroom objects use trigger-based interaction to support collection. The game manager updates the collection counter and checks whether the required number of mushrooms has been collected. Poisonous mushroom scripts apply damage through the player health system.



The health interface displays the current health value. The countdown timer tracks remaining time, and the day-night controller modifies directional lighting. Victory and Game Over panels communicate the final outcome to the player.

IX. Results and Discussion

The developed prototype combines a 3D forest environment with survival-oriented gameplay mechanics. The implemented features include player movement, mushroom collection tracking, poisonous mushroom damage, health management, a countdown timer, a day-night cycle, and victory and Game Over conditions.

Screenshots from the running Unity project can be included to demonstrate the implementation. Actual test results and performance measurements should be recorded after testing the completed build.

X. Limitations and Future Work

The current project focuses on a desktop-based 3D survival game. Future improvements may include additional mushroom types, improved animations, sound effects, inventory features, enhanced environmental assets, and more detailed gameplay statistics.

Augmented Reality and Virtual Reality support could also be explored in future versions. These features would require additional implementation and testing before they could be considered part of the completed system.

XI. Conclusion

The Mushroom Survival Game demonstrates the development of an interactive 3D survival experience using Unity and C#. Players explore a forest, collect safe mushrooms, avoid poisonous mushrooms, manage health, and complete objectives before time runs out.

The project combines character control, object interaction, health management, lighting changes, and game-state handling. It provides a foundation for future improvements in gameplay, graphics, audio, and immersive technologies.

References

[1] Unity Technologies, “Unity Manual.” [Online]. Available: https://docs.unity3d.com/Manual/
[2] Microsoft, “C# Documentation.” [Online]. Available: https://learn.microsoft.com/en-us/dotnet/csharp/
[3] Unity Technologies, “Colliders in Unity.” [Online]. Available: https://docs.unity3d.com/Manual/CollidersOverview.html
[4] Unity Technologies, “Unity UI Toolkit Documentation.” [Online]. Available: https://docs.unity3d.com/Manual/UIElements.html
[5] Unity Technologies, “Animation in Unity.” [Online]. Available: https://docs.unity3d.com/Manual/AnimationSection.html
[6] J. Schell, *The Art of Game Design: A Book of Lenses*, 3rd ed. CRC Press, 2019.
[7] H. Desurvire and M. S. El-Nasr, “Methods for Game User Research: Studying Player Behavior to Enhance Game Design,” *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 82–87, 2013, doi: 10.1109/MCG.2013.61.
[8] L. Caroux, K. Isbister, L. Le Bigot, and N. Vibert, “Player–video game interaction: A systematic review of current concepts,” *Computers in Human Behavior*, vol. 48, pp. 366–381, 2015, doi: 10.1016/j.chb.2015.01.066.
[9] “Assessing the player interaction experiences based on playability,” *Entertainment Computing*, vol. 5, no. 4, pp. 259–267, 2014, doi: 10.1016/j.entcom.2014.08.006.
[10] X. Li, Z. Zhang, and K. Stefanidis, “A Data-Driven Approach for Video Game Playability Analysis Based on Players’ Reviews,” *Information*, vol. 12, no. 3, p. 129, 2021, doi: 10.3390/info12030129.