"Flames of Redemption" Narrative-Driven 2D Platformer Game in Unity Game Engine

Prof. Anuradha S. Deokar [1], Soham R. Shewale[2], Swapnil R. Shardul[3], Amey K.K. Singh[4], Sudipt R. Sharma[5]

[1,2,3,4,5]*Department of Computer Engineering All India Shree Shivaji Memorial Society's College of Engineering, Pune*

ABSTRACT: "Flames of Redemption" is a narrative-driven 2D platformer game developed in Unity, featuring protagonist Felix, endowed with the ability to breathe fire. This research paper explores the technical implementation and narrative design of the game, focusing on algorithms and methods employed to create an immersive player experience. Utilizing Unity's Universal Render Pipeline (URP), Cinemachine for dynamic camera control, and parallax backgrounding, the game sets the stage for Felix's journey through three distinct parts: collecting objects, navigating traps, and confronting a formidable boss. Central to the gameplay are player and enemy physics, diverse projectile algorithms, and UI management, all contributing to a seamless gaming experience. Additionally, the integration of a dialogue system using Ink enriches the narrative, allowing players to engage with characters and shape the story through meaningful choices. The paper delves into the narrative structure, character development, and the pivotal role of player decisions, emphasizing the fusion of technical proficiency with storytelling prowess in game development. Through a detailed analysis of algorithms, dialogue implementation, and narrative intricacies, this paper offers insights into the creative and technical aspects of crafting immersive gameplay experiences.

Keywords: Unity, 2D Platformer, Player Choice, Technical Implementation, Dialogue System, Character Development, Immersive Storytelling, Pathfinding Algorithms, Finite State Machines, Player Choices.

1. INTRODUCTION

In the dynamic landscape of video game development, the convergence of technical ingenuity and captivating storytelling stands as a hallmark of innovation. Amidst this backdrop emerges "Flames of Redemption," a narrative-driven 2D platformer meticulously crafted within the Unity game engine. This paper serves as a comprehensive exploration of the intricate tapestry that intertwines technical implementation with narrative design, illuminating the algorithms, methodologies, and creative processes that underpin the development of this immersive gaming experience. Set in a vibrant world where the protagonist, Felix, possesses the extraordinary ability to breathe fire, "Flames of Redemption" invites players on a journey fraught with peril and possibility. Divided into three distinct parts, the game offers players a multifaceted experience, wherein they must navigate intricate challenges, confront formidable adversaries, and ultimately make pivotal choices that reverberate throughout the narrative fabric.

Central to the gameplay experience is the seamless integration of technical elements, including Unity's Universal Render Pipeline (URP), which facilitates stunning visual aesthetics, and Cinemachine, an innovative tool that empowers dynamic camera control, heightening immersion. Furthermore, the utilization of parallax backgrounding adds depth and dimension to the game world, enriching the player's exploration. At the heart of "Flames of Redemption" lies the intricate dance between player agency and narrative progression. Through the implementation of a dialogue system utilizing Ink, players are afforded the opportunity to engage deeply with the game's characters, shaping the trajectory of the storyline through their choices and actions. As players navigate Felix's journey, they are confronted with moral dilemmas, forging alliances, and facing the consequences of their decisions. As we delve into the technical intricacies of player and enemy physics, diverse projectile algorithms, and UI management, we unravel the underlying framework that drives the gameplay experience forward. Moreover, the paper examines the nuanced nuances of character development and narrative immersion, highlighting the symbiotic relationship between gameplay mechanics and storytelling prowess.

In essence, "Flames of Redemption" represents a testament to the boundless creativity and technical expertise inherent in modern game development. Through a meticulous fusion of technical innovation and narrative depth, the game offers players a compelling and immersive journey, inviting them to explore the power of choice and the depths of redemption within a captivating 2D world.

2. LITERATURE REVIEW

The literature review covers key studies in game development, focusing on Unity, AI algorithms, and agile methodologies. Hussain et al. (2020) emphasize Unity's importance and the need for continuous skill improvement. Nikolaeva et al. (2021) discuss the A* algorithm for pathfinding in Unity games, highlighting design patterns. Sekhavat (2017) surveys behavior trees in popular games for NPC behaviors. Jagdale (2021) explores FSMs and HFSMs, recommending FSMs for simple behaviors and HFSMs for complex ones, with A* for smarter NPCs. McKenzie et al. (2021) examine agile frameworks, suggesting adaptations for game development. Sobota and Pietriková (2023) discuss the educational role of game engines like Unity, Godot, and Unreal Engine. (2021) showcase the Godot Engine's efficiency in creating a 3D puzzle game. Rohini G et al. (2021) highlight Unreal Engine's educational value. Sulaiman and Othman (2023) focus on Model Driven Development and Object Constraint Language in video games, using chess and Silent Hill 2 as models.

3. METHODOLOGY

The development of "Flames of Redemption" as a dynamic 2D game with AI-driven NPCs and a narrative-driven gameplay experience relied on a structured and iterative methodology. The project involved various stages, encompassing design, development, and implementation. The following sections outline the key methodological aspects of this project:

3.1. Project Planning and Conceptualization:
Extensive brainstorming sessions were conducted to define the project scope, goals, and objectives. These sessions served as a platform for the team to explore various ideas and mechanics, aiming to pinpoint concepts that aligned with the project's overarching vision. Through careful evaluation, a final game concept was selected, taking into account factors such as feasibility, engagement potential, and alignment with project goals. Subsequently, a detailed project roadmap was formulated, delineating the development phases, milestones, and deliverables to be achieved throughout the project's lifecycle. This roadmap provided a structured framework that guided the team's efforts, ensuring that development progressed systematically in alignment with the project's objectives.

3.2. Prototyping and Iteration:
An initial prototype was developed to lay the foundation for testing core gameplay mechanics and interactions. Through iterative cycles, the prototype underwent refinements based on feedback from team members, as well as guidance from the project guide and peers. Their insights were instrumental in identifying areas for improvement and enhancing the overall gameplay experience. This collaborative approach ensured that subsequent iterations of the prototype were progressively refined, leading to a more polished and engaging game.

3.3. Art and Asset Selection:
The visual style and aesthetic direction for the game were established through careful deliberation and creative exploration. Concept art, mood boards, and visual references played pivotal roles in guiding the art direction, providing a cohesive visual framework for the game's development. Assets essential to the game's visual identity, including character sprites, environment tiles, animations, and UI elements, were meticulously sourced or created to align with the predetermined aesthetic. Integration of these assets into the game was seamlessly executed, ensuring that they complemented one another and contributed to the overall immersive experience.

3.4. Story and Dialogue Scripting:
The game's narrative, plot, characters, and dialogue underwent extensive development to ensure a compelling and immersive storytelling experience. Dialogue scripts tailored for in-game conversations, cutscenes, and character interactions were meticulously crafted to enrich the narrative depth and player engagement. Integration of dialogue systems was facilitated through the utilization of tools such as the Ink scripting language, enabling the management of branching dialogue paths to accommodate player choices and narrative outcomes.

3.5. Environment Setup and Level Design:
Levels and environments were meticulously designed with a focus on enhancing player progression, providing engaging challenges, and maintaining narrative cohesion. Layouts for platforms, obstacles, collectibles, and interactive elements were carefully crafted to optimize performance while ensuring a seamless gameplay experience. Integration of background elements, lighting effects, and environmental shaders further elevated the atmosphere, immersing players in the game world and enhancing the overall visual appeal and thematic coherence.

3.6. Player and Enemy Mechanics Implementation:
Player movement mechanics were implemented to ensure responsive and intuitive controls, allowing smooth navigation through the game world. Concurrently, enemy AI algorithms were developed to create dynamic and challenging encounters, with enemies exhibiting varied behaviors and attack patterns. Collision detection and response systems were integrated to accurately handle interactions between players, enemies, and environment objects, enhancing the sense of immersion and realism within the game.

3.7. Dialogue System Integration:
Dialogue sequences were seamlessly integrated into the gameplay experience using scripting or dialogue systems, ensuring that narrative elements were delivered effectively throughout the game. Additionally, user interface elements were implemented to display dialogue text, character portraits, and choices, providing players with clear visual feedback and enhancing their engagement with the storyline and characters.

3.8. Trap and Projectile Mechanics Implementation:
Interactive elements like traps, projectiles, and environmental hazards were meticulously designed and implemented to add depth and challenge to the gameplay. Specific algorithms were developed to manage trap activation, projectile behavior, and player interaction within the game environment, ensuring dynamic and engaging gameplay experiences. These elements were seamlessly integrated into the level design, contributing to the overall immersion and excitement of the game.

3.9. Audio Integration:
Carefully selected audio assets, comprising background music, sound effects, and voiceovers, were chosen or crafted to complement the game's atmosphere and enhance the player experience. Advanced audio playback systems were implemented to synchronize sound seamlessly with in-game events and actions, ensuring an immersive and cohesive audio landscape. Through meticulous integration and fine-tuning, the audio elements enriched the gameplay, evoking emotions and enhancing the overall immersion for players.

3.10. Testing and Debugging:
Comprehensive testing was executed across multiple facets of the game to identify and rectify any issues or bugs that may compromise the player experience. Feedback from both peers and the project guide was gathered and meticulously incorporated into subsequent iterations, aiming to enhance stability, functionality, and overall quality. This iterative process ensured that the game underwent continuous improvement, resulting in a polished and refined final product ready for deployment.

3.11. Deployment and Presentation:

Upon completion of development, the game build underwent finalization and preparation for either release or presentation. In tandem, comprehensive documentation encompassing project reports, design documents, and visual aids was meticulously crafted to accompany the game. This documentation served to provide insight into the development process, highlight key features, and offer a comprehensive overview of the project's journey from inception to completion.

By following this methodology, the development of the 2D platformer game was systematically executed, ensuring that each step contributed to the overall success and quality of the final product.

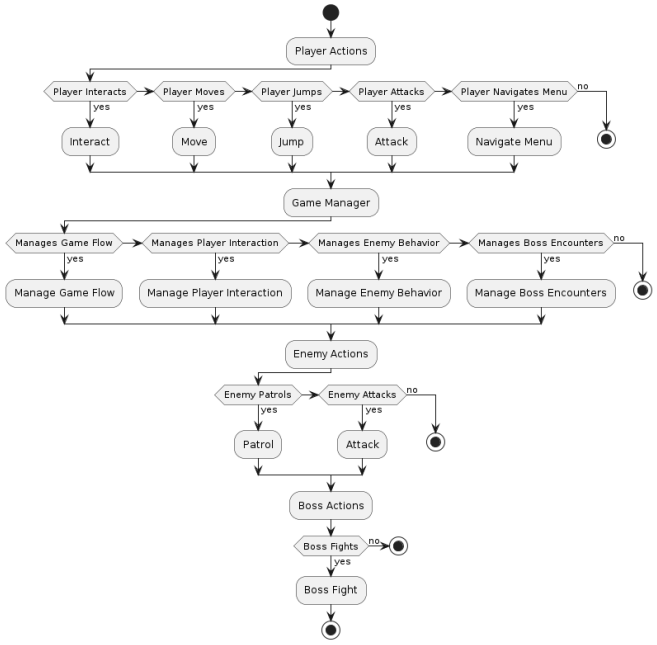

Fig 3.1. Flow Diagram

4. ALGORITHMS FOR NPC BEHAVIORS

The core of "Flames of Redemption" lies in the life like behaviors of its non-playable characters (NPCs), each brought to life through specific AI algorithms tailored to their roles. These algorithms have been instrumental in enriching gameplay and enhancing player engagement.

4.1. Grunt Enemies:

Pathfinding Algorithm (A* Algorithm): Grunt Enemies employ the A* algorithm to track and attack the player strategically. By utilizing this advanced pathfinding technique, Grunt Enemies can dynamically navigate the game world, seeking out the player's position while evading obstacles. The A* algorithm enables Grunt Enemies to plan efficient routes to intercept the player, contributing to their aggressive and challenging behavior in combat scenarios.

Fig 4.1. Implementation of A* Pathfinding

4.2. Player Mechanics:

Finite State Machine (FSM): The player character's behavior is managed by a Finite State Machine (FSM), which controls actions like walking, jumping, attacking, and interacting. This FSM allows for smooth transitions between states based on player input and environmental factors, ensuring responsive controls and dynamic gameplay. Additionally, animations are synchronized with each state, enhancing immersion, while state-based interactions enable players to engage with various elements in the game world.

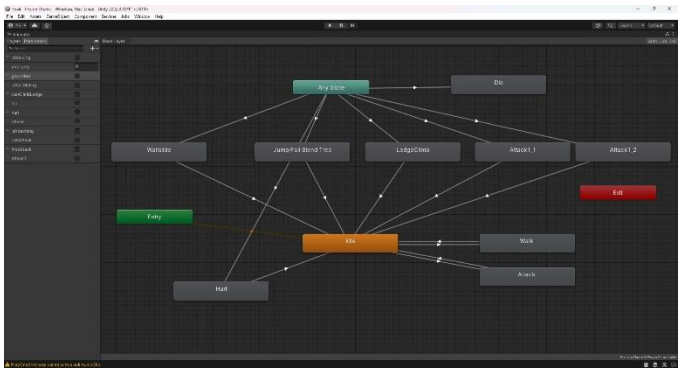

Fig 4.2. Finite State Machine of Player

4.3. Boss AI (NPC):

Behavior Trees: Boss Morbius employs Behavior Trees to dictate its actions and decisions during gameplay. These trees consist of sequence nodes for sequential actions, selector nodes for decision-making, composite nodes for complex behaviors, and decorator nodes for conditional modifications. This architecture allows Morbius to exhibit dynamic and challenging behavior, enhancing the boss battle experience for players.

5. RESULT

The development of our 2D platformer game, titled “Flames of Redemption”, culminated in a fully functional and engaging gaming experience. Through the implementation of various algorithms, tools, and design principles, we were able to achieve our project objectives effectively. Below, we outline the key results of our project:

5.1. Gameplay Mechanics:

The protagonist's ability to breathe fire added a unique gameplay element, allowing for both offensive attacks and puzzle-solving mechanics. Utilizing behavior trees and A*

pathfinding, the enemy AI exhibited sophisticated behaviors such as patrol routes, pursuit of the player, and evasion of hazards, enhancing the challenge and immersion of gameplay. The incorporation of traps and boss fights in different parts of the game added diversity and excitement to the gameplay experience.

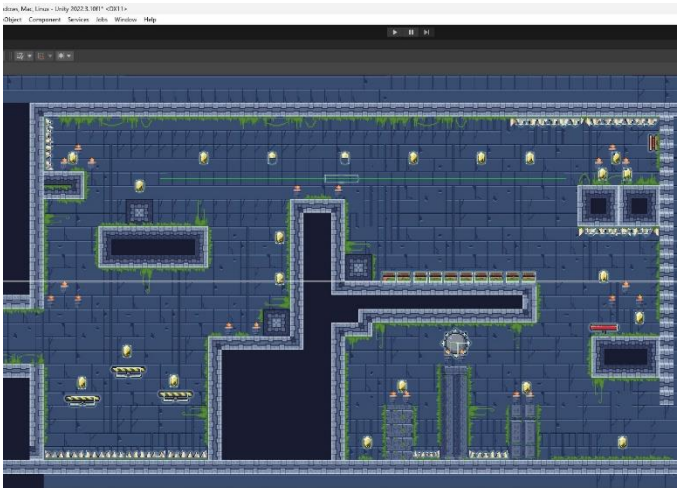

Fig 5.1. Trap Level

5.2. Technical Implementation:
Utilizing the Unity Universal Render Pipeline enabled efficient rendering and optimized performance without compromising visual quality. Integration of the Cinemachine tool facilitated dynamic camera movements and effects, enhancing immersion during gameplay. Parallax scrolling backgrounds added depth to the game's environments, enriching the visual experience. Implementation of realistic physics for player and enemy interactions ensured smooth and responsive gameplay mechanics.

5.3. Storytelling and Player Choice:
The use of Ink for dialogue implementation allowed for branching narratives and player choices, enhancing player engagement and immersion. The three-part structure of the game, coupled with branching storylines and multiple endings, provided players with meaningful choices and a sense of agency.

Fig 5.2. Ink Dialogue Implementation

5.4. Audiovisual Elements:
Integration of appropriate music and sound effects enhanced the atmosphere and immersion, contributing to the overall polish of the game. Effective UI design and text management provided clear feedback and guidance to players, enhancing the user experience.

Fig 5.3. Audio Integration

5.5. Overall Experience:
Playtesting feedback indicated high levels of engagement and enjoyment among players, with positive remarks regarding gameplay mechanics, storytelling, and visual/audio elements. The branching storyline and multiple endings encouraged replayability, allowing players to explore different narrative paths and outcomes.

Fig 5.4. Endgame Screen

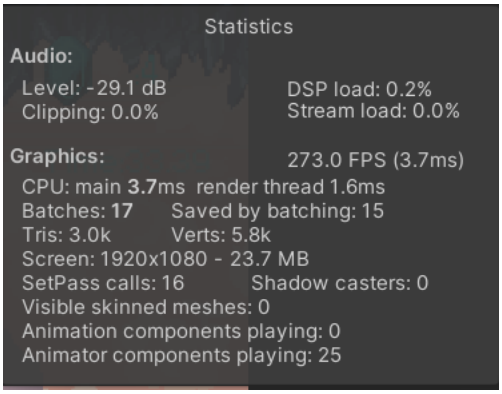

Fig 5.5. Final Game Statistics

The profiler stats indicate that the game is running very efficiently, with high FPS, low CPU and render thread times, and effective batching. There are no significant audio or graphical performance issues. The absence of shadows and skinned meshes simplifies rendering, contributing to the high-performance metrics. These insights are critical for optimizing and maintaining a smooth gameplay experience. The high FPS and low processing times suggest that the game can handle additional complexity or higher resolution assets if needed.

6. CONCLUSION

In conclusion, the development of *Flames of Redemption* represents a significant advancement in the field of game development, showcasing the successful integration of various algorithms, tools, and design principles to create an immersive 2D platformer game.

Through meticulous implementation, the game features unique mechanics such as the protagonist's fire-breathing ability and sophisticated enemy AI with behavior trees and A* pathfinding. These elements contribute to a dynamic gameplay experience, offering players challenging obstacles and engaging interactions.

Looking ahead, there are numerous opportunities for future expansion and enhancement of *Flames of Redemption*. Potential avenues for further development include the addition of new levels, bosses, enemies, traps, and narrative arcs to provide players with varied and compelling gameplay experiences. Additionally, ongoing updates focused on refining gameplay balance and incorporating player feedback could further improve the overall gaming experience. The successful development of “Flames of Redemption” highlights the potential for interdisciplinary collaboration between computer science and game design disciplines. By leveraging advanced algorithms and technical expertise, developers can create immersive and engaging gaming experiences that push the boundaries of interactive entertainment.

As the gaming industry continues to evolve, projects like “Flames of Redemption” serve as valuable contributions to the growing body of knowledge in game development. By applying scientific principles and methodologies to game design, developers can continue to innovate and create experiences that captivate and inspire players worldwide.

In closing, “Flames of Redemption” stands as a testament to the creativity, technical skill, and collaborative effort of the development team. We look forward to the continued evolution of the game and the opportunity to further explore the possibilities of game development in future projects.

7. REFERENCES

1. Hussain, Afzal & Shakeel, Haad & Hussain, Faizan & Uddin, Nasir & Ghouri, Turab. (2020). Unity Game Development Engine: A Technical Survey. University of Sindh Journal of Information and Communication Technology. 4.

2. D. Nikolaeva, M. Safi, M. Mihailov, A. Georgiev, V. Bozhikova and M. Stoeva, "Algorithm A* and Design Patterns used in Unity Video Game development," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-3, doi: 10.1109/ICAI50593.2020.9311327.

3. Sekhavat, Yoones. (2017). Behavior Trees for Computer Games. International Journal on Artificial Intelligence Tools. 26. 10.1142/S0218213017300010.

4. Jagdale, Devang. (2021). Finite State Machine in Game Development. 384-390. 10.48175/IJARSCT-2062.

5. C. Tang, Z. Wang, X. Sima and L. Zhang, "Research on Artificial Intelligence Algorithm and Its Application in Games," 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), Manchester, United Kingdom, 2020, pp. 386-389, doi: 10.1109/AIAM50918.2020.00085.

6. T. McKenzie, M. Morales-Trujillo, S. Lukosch and S. Hoermann, "Is Agile Not Agile Enough? A Study on How Agile is Applied and Misapplied in the Video Game Development Industry," 2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE), Madrid, Spain, 2021, pp. 94-105, doi: 10.1109/ICSSP-ICGSE52873.2021.00019.

7. M. Cook, "Software Engineering For Automated Game Design," 2020 IEEE Conference on Games (CoG), Osaka, Japan, 2020, pp. 487-494, doi: 10.1109/CoG47356.2020.9231750.

8. E. Annanperä, J. Yli-Kantola, T. Sauvola, S. Heinonen and E. Siira, "Testing methods for mobile game development a case study on user feedback in different development phases," 2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH), Vienna, Austria, 2018, pp. 1-8, doi: 10.1109/SeGAH.2018.8401372.

9. Game Development Using Unreal Engine 4, 2021, https://doi.org/10.26483/ijarcs.v12i0.6722, Published: Aug 3, 2021

10. 3d Puzzle Game Using Godot Engine, 2021, https://www.ijedr.org/papers/IJEDR2102023.pdf, Published: 2021

11. Kerr, Aphra. (2021). Before and After: Towards Inclusive Production Studies, Theories, and Methods. 10.5117/9789463725439_After.

12. V. T. Sarinho and A. L. Apolinário, "A Generative Programming Approach for Game Development," 2009 VIII Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro, Brazil, 2009, pp. 83-92, doi: 10.1109/SBGAMES.2009.18.

13. El Rhalibi, A. & Wong, Kok & Price, Marc. (2009). Artificial Intelligence for Computer Games. Int. J. Computer Games Technology. 2009. 10.1155/2009/251652.

14. Dubois, Louis-Etienne & Weststar, Johanna. (2021). Games-as-a-service: Conflicted identities on the new front-line of video game development. New media & society. 24.

2332-2353. 10.1177/1461444821995815.

15. Phuong, Nguyen & Nguyen, Luong. (2021). A study on game consumer behavior. Management Science Letters. 11. 2323-2330. 10.5267/j.msl.2021.6.002.

16. Zeiler, X., & Mukherjee, S. (2022). Video Game Development in India: A Cultural and Creative Industry Embracing Regional Cultural Heritage(s). Games and Culture,17(4),509-527. https://doi.org/10.1177/15554120211045143

17. Manzano, Clemente & Lermanda, Tomás & Martínez-Araneda, Claudia & Vidal, Christian & Segura, Alejandra. (2022). Teach me to play, gamer! Imitative learning in computer games via linguistic description of complex phenomena and decision trees. Soft Computing. 27. 10.1007/s00500-022-07476-z.

18. Engström, Henrik. (2020). Game Development Research.

19. Sulaiman, Sazan & Nuree, Rawshan. (2023). Object Constraint Language for Modelling Computer games.

20. Uludağ, Kadir. (2023). Crafting Interesting Storytelling Games with ChatGPT: Unleash Your Imagination.

21. Mileff, Peter. (2023). PRACTICAL GUIDE TO IMPLEMENT A SIMPLE 2D GAME ENGINE. 11. 27-50. 10.32968/psaie.2023.3.3.

22. Kishor, Kaushal & Rani, Rupa & Rai, Atul & Sharma, Varsha. (2023). 3D Application Development Using Unity Real Time Platform. 10.1007/978-981-99-3716-5_54.

23. Sobota, Branislav & Pietriková, Emília. (2023). The Role of Game Engines in Game Development and Teaching. 10.5772/intechopen.1002257.