**From Concept to Code: Shadows of the Forsaken**

**A Developer’s Perspective**

Course Code & Section: CSE 115.2

Project Group 1

Names & IDs

Md. Fahim Hossain, Samia Zaman, Sidratul Muntahara Audhira, Md. Lutful Alam  
2512847642, 2511026642, 2513009642, 2412032042

# ***Abstract****—* Game development is a complex amalgamation of creativity, technical proficiency, and problem-solving skills. "From Concept to Code: Shadows of the Forsaken—A Developer’s Perspective" encompasses the creation, development, and refinement of a text-based adventure game in which players navigate an immersive world influenced by their choices. This project brings back the most important parts of interactive storytelling in a time when graphics are very advanced and rendering happens in real time. It shows that even complex graphics are not the primary consideration, since engaging storylines and strategy choices may improve the enjoyment of a game. This document puts together all of the steps of the process, including planning and research, design and implementation, alongside the most important details and challenges encountered during the process. The review of the literature looks into the history and evolution of text-based games and analyses their relevance juxtaposed to contemporary interactive fiction. This research examines narrative patterns, methods of player interaction, and mechanisms for player decision-making. It shows why text-based adventures are still popular and how they can immerse players through language alone. The methodology section outlines the fundamental elements of the game's architecture, including problem description, core mechanics, and system design. The document outlines the game concept and narrative, carefully crafted to evoke surprise, challenge, and emotional engagement. This part analyses the fundamental code architecture, encompassing essential components such as NPC behaviour, decision trees, user input and output management, game loops, logical flow, and save/load functionalities. The game is built in C, employing structured programming and efficient algorithms to ensure smooth performance and user involvement. The system requirements are examined, providing insights into the employed libraries, font selections, and the tools necessary for the efficient development and execution of the game. This study candidly addresses the challenges and constraints encountered during the development process, acknowledging the inherent barriers involved. Debugging complex conditional branching and enhancing user experience and interaction mechanisms presented opportunities for learning and development at every challenge. Enhancing game performance, addressing edge-case scenarios, and maintaining a seamless decision-driven narrative flow were some of the most challenging challenges undertaken throughout development. The existing implementation provides a comprehensive and engaging experience, while the scope for future improvements is substantial. The report describes planned changes that will happen, such as adding a graphical user interface (GUI) to make the game easier to access and use, a multiplayer mode so that players can work together or compete, and smart non-playable characters (NPCs) that can make decisions based on how players interact with them. These innovations would extend the boundaries of text-based games, linking traditional interactive fiction with modern AI-driven storytelling. The provided appendices illustrate the game while offering code snippets, thus enhancing one's understanding of the logic and mechanics of its execution. The comprehensive reference list and bibliography grant credit to the books, journals, research papers, and other internet sources that served as the project’s theoretical framework and technology execution documentation. “Shadows of the Forsaken” is more than a mere game; it is a perfect example of the effectiveness of text-based storytelling and the complexity that goes into game development. This document is a developer’s diary on how one’s idea transformed into an operational interactive experience. This document is important for novice game developers, avid readers of interactive fiction, and programmers by showing how the fusion of complex narrative, a decision-making system, and programming logic resulted in a captivating and fascinating game. This project demonstrates how far the world of interactive fiction can go with the combination of an artistic mind and technical skill; it demonstrates that the greatest stories rarely are found on the screens but within our minds. Also this report examines various motivation sets, including their categories and roles, the design of motivation for implementation, and possible future adjustments to gamification to enhance user engagement.

# ***Keywords: text based adventure game, game narrative. interactive storytelling, branching narrative, dynamic content generation, psychological thriller, immersive experience, C programming, game design, player engagement.***

# **INTRODUCTION**

Every decision bears grave consequences. "Shadows of the Forsaken," a blend of horror and adventure, immerses players in a chilling, text-based thriller game where every choice could lead to their doom. Shadows of the Forsaken is more than just a game; it's an immersive experience that immerses players in a terrifying and uncertain world. This nightmarish combination of adventure and horror plunges users into a dark text-based thriller where every battle aims to claim their life and every ounce of blood turns the story deeper and deeper, fogging the boundary between breath into whiff and death into doom. The game begins, and the narrative unfolds as follows: The first chapter begins with a mundane adventure dominated by a silencing misty forest, where the trees can be likened to mass living beings and where in reality everything humanoid captures the mind, and a careful observation makes everything otherworldly. Players become more immobile the higher and bolder they ascend, raising the question: are they truly lost? Every aspect of the world is manipulating them as they steadily navigate toward reality. Eventually, an ominous squeezed mansion emerges from beyond the mist—pruned along its edges. Try as the players might to get away, they fail because reality collectively decouples their will. It is as though some malevolent deity has complete dominion, scripting their tale beforehand, granting them the illusion of autonomy fraught alongside the matter of harrowing control. As you explore the deepest parts of the forest, it becomes clear that something evil is pulling you toward a scary house. You will always end up in this empty, scary place, even if you try to get away. It's like the forest has a mind. The console commands are simple, but every time you press them, you'll find something horrible. This is similar to how the story alternates between hope and despair, leaving you questioning your ability to escape.

# **LITERATURE REVIEW**

Interactive storytelling has been an essential component of human civilization for ages from oral tradition to the initial manifestations of written storytelling. Interactive storytelling games are games where the player participates in the storytelling process in any way that they can imagine, and the game attempts to create a coherent dramatic experience around their input. Only since the early 2000s have they started to manage anything close. For Chris Crawford and scholars like Bostan and Marsh, and for the context of this dissertation, Interactive storytelling is not interactive fiction, or interactive narrative. Interactive storytelling does not use a branching structure to organize its content, but organizes and generates that content dynamically based on interaction (Miller, 2021). In the digital era, this narrative heritage evolved into text-based games, wherein the player's imagination compensates for the omissions inherent in the text.These games provide a special kind of involvement by combining interactivity with reading and making players negotiate decisions that affect the course of the story. With an eye toward how they still enthrall players despite the predominance of high-resolution graphics and theatrical gameplay, this part investigates the beginnings, development, and current scene of text-based adventures.

## **Overview of Text Based Games:** One of the earliest forms of gaming is the text-based or interactive fiction (IF) games, where users provide input and control everything through imagination and creativity instead of using monitors. Their focus is the intricately woven story coupled with decision-making. Unlike today's video games loaded with graphics and animations, these games are simplistic, combining a story and the user's imagination. The "choose your own adventure" style of the most primitive text-based video games originated from board role-playing games like Dungeons and Dragons. These games engage players by requiring them to type out commands to progress, like "open door" or "attack monster." What was most enchanting about the games was the level of control each player was given and how their choices impacted the story. The heart of these types of games lies in the strong narrative, which is prioritized over flashy mechanics and graphics. The players are downloaded into a world crafted with text, giving them a sense of weight for their choices that even the most visually stunning games fail to achieve.

1. **Evolution of Text-Based Games:**

The evolution of text-based games, from being trivial experiments to core parts of gaming, is truly something admirable. Will Crowther developed a computerized version of a dungeon puzzle game called “Colossal Cave Adventure” back in 1976. This game featured fascinating caves and puzzles and had an interactive storyline unlike anything that had come out previously. This game single handedly changed the entire concept of interactive fiction. Zork built upon 'Colossal Cave Adventure,' adding humor and even more vividly described worlds. Players in 1980 could find themselves in the remains of the grand underground empire and face terrible creatures, called grue, that waited in the dark to capture any carefree traveler. The game also introduced richly described settings and advanced techniques to craft new items from old ones. During the 1980s and 1990s, Infocom, the creators of The Hitchhiker’s Guide to the Galaxy and Deadline, revolutionized text-based games by popularizing their use. These games allowed for advanced command parsers to carry out multi-step instruction sequences instead of simple noun-verb commands. As technology advanced, it led to an increased emphasis on graphics. This led to the decline of text-based games and the rise of point-and-click, RPG, and 3D open-world games. Even though these new games heavily relied on visual aspects, text-based games still survived and evolved, proving their worth.

## **Existing Similar Games (Related Works):** Numerous games have tried to capture the unsettling and chilling atmosphere that Shadows of the Forsaken aims to achieve. The horror genre, in particular, has been fertile ground for text-based experiences, using psychological terror alongside gripping narratives to capture the audience's attention. Here are some of the most influential titles related to our venture.

* **Anchorhead (1998)** Anchorhead is the quintessential embodiment of a Lovecraftian horror. It follows the tale of a couple relocating to a spooky seaside town known for some extremely dark secrets. Unequivocally, the game combines atmospheric descriptions and elaborate puzzle games, along with a slow but constantly developing sense of dread, which makes it arguably one of the best interactive horror experiences of all time.
* **Stories Untold (2017)** Horror has found its new digital form in Stories Untold. The game integrates elements of visual and audio features along with writing to give the players a different type of horror experience. The opening chapter, The House Abandon, is frightening, utilizing nostalgia while subverting player expectations in terrifying ways.  
   These games show the fact that text-based storytelling can ensnare the gamer’s complete attention, especially in a horror-themed game. The diversification of gaming technology amplifies the advancement in horror storytelling through words. Frightening is the story where all illustrations are the only terrifying thing that can come in the form of a sole screen telling the entire plot, prompting the brain to conceive bleakness even more disturbing than anything a screen could give life to.

# **METHODOLOGY**

When designing Shadows of the Forsaken, we focused on the building blocks of how the game looked a lot like an RPG and a command-line text-based game merged into one with the stories we created. The game was written entirely in C, using only text I/O to represent the ambient setting, challenges, and player progress. Here were the steps of our development process:  
Created a branching plotline with multiple decision points for replayability & psychological tension. Utilized conditionals and loops to design game states, decision trees and player input handling mechanisms.We designed the game to be made up of separate modules for environment rendering, command parsing, event triggers, and inventory management. Continually test for bugs in-game and logic errors while improving the atmosphere, pacing, and emotional attachment through player feedback. Streamlined UI provided succinct interactions when necessary, communicating through text prompts, timing pauses, and descriptive prose to maintain tension.

## **PROBLEM DEFINITION**

In the context of contemporary gaming, characterized by hyper-realistic graphics and rapid gameplay dynamics, a significant challenge arises in the development of a compelling horror-adventure experience that relies exclusively on textual narratives. The objective centers on extending the limits of imagination and enhancing player immersion solely through narrative constructs and the aspect of choice. To achieve this aim, several critical elements must be addressed:  
Firstly, it is essential to maintain psychological tension and unpredictability throughout the experience. This entails crafting scenarios that elicit fear and suspense without reliance on visual stimuli. In tandem with this requirement, the simulation of environmental exploration and unsettling encounters must be effectively communicated through descriptive language that evokes a vivid sense of place and atmosphere.  
Furthermore, the design must prioritize a user interface that allows for smooth and responsive interaction via basic commands, thereby facilitating player engagement with the narrative. Additionally, the story must be constructed in a manner that dynamically responds to player choices, ensuring that the unfolding narrative reflects the decisions made, thus enhancing the overall sense of agency.  
Ultimately, the overarching aim is to devise a text-based horror system that conveys a sense of vitality and manipulation. This system should instill in players a feeling of diminished control,   
despite the fact that they are the ones making decisions. Through this approach, the intention is to create a psychological experience that transcends traditional gaming modalities, challenging players to navigate their fears through the power of words alone.

### **GAME CONCEPT & STORYLINE**

The psychological horror game, Shadows of the Forsaken, is presented in a text-based adventure format, situated within an ever-evolving forest characterized by both vitality and malevolence. Players are immersed in a grim, mist-laden environment from the outset of their journey, which initially appears to be routine; however, they soon become aware of an unseen force that exerts control over their actions.

The narrative is structured into distinct chapters, commencing in a fog-drenched forest where each tree observes with an unsettling awareness and every auditory cue resonates with an atmosphere of dread. As players traverse deeper into this treacherous landscape, encounters with peculiar markings, distorted pathways, and ominous visions emerge. Regardless of the players' decisions, an inexorable progression toward an ancient mansion, which looms ominously in the background, is experienced; its influence intensifies with each step taken within the forest.  
  
At the heart of this narrative lies the theme of manipulation and control, prompting a fundamental question regarding the player's autonomy: Is there genuine freedom, or is the individual merely a pawn ensnared within a predetermined nightmare? The forest and the mansion serve not only as mere backdrops but are integral characters that influence and mold the trajectory of the player's experience.

### **KEY FEATURES**

1. Narrative Driven by Choices: Each decision made by the player influences the storyline, resulting in various outcomes and concealed narrative branches.
2. Text-Based Horror Setting: The game employs richly detailed descriptions to fully engage the player in a landscape characterized by pervasive dread, obviating the necessity for visual elements.
3. Psychological Tension: The experience is anchored in cultivating a sense of unease through effective storytelling, deliberate pacing, and the illusion of player autonomy.
4. Diverse Endings: The fate of the player is contingent upon their actions, resulting in scenarios such as escape, entrapment, or more severe consequences.
5. User-Friendly Command Inputs: The gameplay is made accessible through typed commands, allowing players to concentrate on the narrative and atmospheric elements.
6. Replay Value: Each playthrough reveals new pathways and insights, motivating players to investigate every available choice.
7. Adaptive Story Events: The narrative subtly adjusts in response to player behavior, ensuring an unpredictable and continually evolving experience.

### **CODE ARCHITECTURE**

The game is developed using modular C functions, with a clear separation between logic, navigation, and storytelling. Each gameplay component is encapsulated within its own function, promoting readability and future scalability. The main() function initializes the main menu, and scene progression is handled through function calls such as new\_game(), abandoned\_mansion(), and game\_over(). A switch-based system and conditionals govern user interaction and path branching, ensuring flow consistency and decision-driven outcomes.

### **NPC (Non-Playable Characters):** Though text-based, *Shadows of the Forsaken* integrates NPCs like **Fahim**, **Samia**, **Audhira**, **Jannat**, and **Lutful** to guide narrative development. Each NPC contributes to tension and depth in the story, representing different player perspectives (fear, curiosity, rationality). They don’t act independently but influence the story through dialogue and emotional cues, enhancing immersion and setting the mood for critical decisions.

### **Decision Tree:** The core gameplay is built around a **decision tree structure**, where each choice leads to unique outcomes. For instance: A. Choosing to *enter the mansion* progresses the story. B. Choosing to *stay outside* leads to a side event and *game over*. C. Other scenes will expand based on future sceneNo values and branching logic. D. This branching mechanism is controlled using conditionals (if-else) and switch-case blocks, directing the player’s path based on their input.

### **Handling Text (Input/Output):** All game interactions are handled using **console input and output**: A. printf() is used to narrate the story and present choices. B. getch() and scanf() handle player responses. C. A custom function get\_choice() ensures valid input and listens for special keys like ESC (to trigger the menu) and 'm' (to display the map). This structure ensures minimal errors and maintains a smooth flow for the player.

### **Game Loop & Logic Handling:** The game doesn’t use a traditional while loop but relies on **recursive function calls** and branching to simulate a game loop. The player’s journey is tracked using the sceneNo, and logic flow is managed through: A. Function calls (new\_game(), saveVault(), menu()). B. Conditional paths for each scenario. C. Reusable user input management through get\_choice(). Each “scene” is a logical block that either advances the story or ends it, mimicking a state machine architecture.

### **Save & Load:** The game implements a basic but functional save/load system: **A. Save**:The save(int sceneNo) function writes the current scene number to a load.txt file. **B. Load**: The load() function reads from load.txt and uses saveVault(sceneNo) to resume the game. The system is file-based, offering persistent session management between plays and allowing players to pick up from where they left off.

### 

### **PROBLEM FACED & DEBUGGING**

Several challenges arose during development:  
**1. Input Handling:** Preventing crashes from invalid input required refining get\_choice() to manage unexpected key presses.  
**2. File Handling:** Ensuring save/load operations worked consistently across platforms required careful handling of file pointers.  
**3. Flow Disruption**: Recursive calls to main() and other functions had to be managed to avoid stack overflow and ensure smooth backtracking.  
**4.Scene Management**: Creating a scalable system for scenes led to the use of scene numbers and a switch-style saveVault() dispatcher.All issues were debugged using printf()-based tracing and logical testing.  
  
  **SYSTEM REQUIREMENTS  
1. Operating System**: Windows 7 or higher (for <conio.h>)  
**2. Compiler**: GCC or Code::Blocks  
**3.** **Memory**: Minimal (~10 MB)  
**4. Storage**: Requires less than 1 MB (only a text file for save data)  
**5.Console**: Any standard terminal or CMD interface  
  
 **LIBRARIES USED  
1. <stdio.h>:** Standard input/output for story narration and text display.  
**2. <string.h>:** Reserved for future string manipulation, e.g., comparing answers in riddles.  
**3. <stdlib.h>:** Used for the exit() function and memory management.  
**4. <conio.h>:** Enables use of getch() to detect special keys like ESC or character commands.  
  **FONT**As a **console-based text game**, the visual experience heavily depends on the **system font of the terminal**. It is recommended to use:  
**1. Font**: Consolas or Courier New (monospaced for formatting consistency)  
**2.Size**: 12–14pt  
**3.Color Scheme**: Dark background with light text (for better readability and atmosphere).Customizing the terminal font and color can significantly enhance immersion.  
  
 **CHALLENGES & LIMITATIONS**Despite the success in building a functional and atmospheric experience, the project faced several challenges:  
**1. Limited by Text Only**: Conveying fear and tension without visuals or sound required heavy reliance on descriptive writing and player imagination.  
**2.Scalability**: As more scenes were added, organizing and maintaining them became harder without a scene management system or scripting engine.  
**3.Console Constraints**: Lack of advanced user interface capabilities (animations, colored text, sound) due to console limitations.  
**4.No Error Handling in File I/O**: Early stages lacked robust error handling for corrupted or missing save files.  
**5.Single-Threaded Flow**: The current system is linear and synchronous, making it hard to incorporate background events or time-based triggers  
   
 **FUTURE WORKS**We envision several improvements to enhance gameplay and player engagement in future versions:  
**1. GUI Implementation**: A graphical interface to replace the console environment, improving accessibility and visual impact.  
**2.Expanded Narrative Branches**: More detailed decision trees, including personality-based consequences.  
**3.Dynamic NPC Behavior**: Making NPCs react to player decisions, develop over time, or even betray or assist the player.  
**4.Audio & Visual Feedback**: Adding background soundtracks, visual effects, and jumpscares to elevate the horror element.  
**5.Inventory System**: Introducing object collection, usage, and puzzle-solving mechanics.  
**6.Mobile & Web Porting**: Adapting the game for cross-platform availability.

### **GUI INTERFACE** A future GUI version of *Shadows of the Forsaken* would: **1.** Replace text commands with clickable choices. **2.** Include visual representations of scenes, rooms, and NPCs. **3.**Provide a sidebar with player inventory, clues, and map. **4**.Use animation for transitions, lighting changes, and effects. **5**.Offer mouse and keyboard input modes. This would significantly enhance immersion and attract a wider audience, especially players unfamiliar with text-based interfaces **MULTIPLAYER MODE** Though the current version is single-player, a future multiplayer mode could introduce: **1. Co-op Gameplay**: Players control different characters with unique perspectives and split decision paths. **2. Shared Puzzle Solving**: Requiring real-time collaboration to solve in-game mysteries or avoid death. **3. Role-based Progression**: Each player may receive different clues or information, encouraging communication and teamwork. This could be developed over a network or in a turn-based format to maintain narrative cohesion. **SMART NPC INTERACTION** A proposed upgrade involves **intelligent NPCs** that: 1. React to prior choices (trust, fear, suspicion). 2. Help or hinder based on player alignment or past decisions. 3. Offer dialogue choices that influence relationships. 4. Learn from player behavior and adapt responses (basic AI). **APPENDICES**

#### **Appendix A: Scene Naming Convention** Each major room or moment is tied to a unique scene number (used in sceneNo variable) for saving/loading and internal mapping.

#### **Appendix B: Command Shortcuts 1.** Press ESC: Open in-game menu **2.** Press M: View map **3.** Numeric keys: Make decisions (e.g., 1, 2, 3)

#### **Appendix C: Project Files 1.** main.c – Source code file **2.** load.txt – Save file **3.** README.txt – Setup & instructions (optional)

### **CODE SNIPPETS** Here are a few notable snippets from the game code:

// Input Handler with Escape and Map Key Support

int get\_choice(int num\_choices, int sceneNo) {

char input;

while (1) {

input = getch();

if (input == 27) menu(sceneNo);

if (input == 'm' || input == 'M') map();

if (input >= '1' && input <= '9') {

int choice = input - '0';

if (choice >= 1 && choice <= num\_choices) return choice;

}

printf("\nInvalid choice. Please try again.\n");

}

}

// Save Game Functionality

void save(int sceneNo){

FILE \*file = fopen("load.txt", "w");

if (file == NULL) {

printf("Error saving file!\n");

return;

}

fprintf(file, "%d", sceneNo);

fclose(file);

printf("Saved Successfully!!\n");

menu(sceneNo);}

// Prologue Narrative

printf("\"Some places are meant to be forgotten... but they never forget you.\"\n");

printf("Deep within the heart of an ancient forest stands a mansion that time has abandoned...\n");

int main(void){

printf("Welcome to {Game Name}");

printf("\n1. NEW GAME");

printf("\n2. CREDITS");

printf("\n3. EXIT");

int choice = get\_choice(3);

if (choice == 1) {

new\_game();

}

else if (choice == 2) {

credits();

}

else if (choice == 3) {

printf("\nAre you sure you want to exit?");

printf("\n 1. Yes :( 2. No :D");

int exit\_choice = get\_choice(2);

if (exit\_choice == 1){

printf("Exiting the Game.....");

exit(0);

} else if (exit\_choice == 2){

main(); // Restarts the menu

}

}

return 0;

}

### **WALKTHROUGH**

**Beginning:  
1.** Start the game and select "New Game".  
**2**.Read through the prologue and introductory dialogues with your friends.  
  
**First Choice:  
1. Enter the mansion** → Progress story into the main game.  
**2. Stay outside** → Leads to mysterious forest collapse and ends the game.  
**3. Knock on the door** → Leads to an unexpected grab and immediate game over.  
  
**Tips:  
1.** Press **M** to view the map when unsure where you are.  
**2.** Press **ESC** to access the in-game menu and save your progress.  
**3.** Read carefully—many clues and hints are hidden in the text.  
**4.** Every decision matters—wrong choices often lead to death.

**CONCLUSION**

Shadows of the Forsaken is much more than a typical text-based game. It's a psychological journey through fear, doubt, and the perception of control. Made with passion and intent, the game weaves together narrative, choice, and tension to deliver an intensely immersive experience within the limitations of a console interface. Each line of code is there to build atmosphere, and each player choice nudges the narrative toward salvation or destruction.

From the foreboding prologue to the branching endings, the game is an example of how creativity and storytelling can thrive within minimalistic design. Dynamic scene transitions, save/load functionality, environmental descriptions, and branching decisions were all essential components that were implemented to craft a coherent and engaging game framework. Problems along the way were solved with innovative workarounds, and the technical limitations were transformed into means of fostering imagination and player immersion.

Looking to the future, Shadows of the Forsaken holds immense potential for growth. The implementation of a graphical user interface, NPC artificial intelligence, and multiplayer capability will not only enhance its usability but unlock new levels of interactivity and replayability. With its solid groundwork and engaging core, this project stands as a testament to the strength of narrative-driven design—and its ability to extend far beyond the console screen.

In total, it's not just a survival game; it's about facing the unknown, challenging perception, and surviving in a world where even reality is a mystery. It's a chilling reminder: there are some places that are not meant to be remembered… but they never forget you.

**REFERENCES:**

* Miller, E. S. (2021). TOWARD THE DESIGN OF INTERACTIVE STORYTELLING GAMES THAT TEACH COMPUTATIONAL THINKING. 270. <https://utd-ir.tdl.org/server/api/core/bitstreams/fb04be69-f03f-4117-b05e-1fd4d0e325f3/content>
* Anchorhead. (2024, December 19 ). In *Wikipedia.*

<https://en.wikipedia.org/w/index.php?title=Anchorhead&oldid=1264000527>

* Stories untold. (2017, February 27 ). In *Wikipedia.*

<http://en.wikipedia.org/w/index.php?title=Stories_Untold_(video_game)&oldid=1258382239>

* Montfort, N. (2005). *Twisty Little Passages: An Approach to Interactive Fiction*.
* Meehan, J. R. (1977). *TALE-SPIN, An Interactive Program that Writes Stories*.
* Lebowitz, J., & Klug, C. (2011). *Interactive Storytelling for Video Games: A Player-Centered Approach to Creating Memorable Characters and Stories*.
* Short, E. (2019). *Interactive Fiction Community Forum & Inform Blog*.  
  <https://emshort.blog/>
* Wikipedia contributors. (2024, March 10). *Zork*. In Wikipedia.  
  <https://en.wikipedia.org/wiki/Zork>
* Wikipedia contributors. (2024, March 10). *Zork*. In Wikipedia.  
  <https://en.wikipedia.org/wiki/Zork>
* Preece, J., Rogers, Y., & Sharp, H. (2015). *Interaction Design: Beyond Human-Computer Interaction* (4th ed.).  
   Wiley.
* Unity Technologies. (2023). *Building narrative-driven games*. Unity Learn.  
   <https://learn.unity.com/tutorial/building-narrative-driven->
* OpenAI. (2023). *GPT-4 Technical Report*.<https://openai.com/research/gpt-4>
* -Bell, M. (2021). *Game design deep dive: Horror*. Game Developer.  
   https://www.gamedeveloper.com/design/game-design-deep-dive-horror