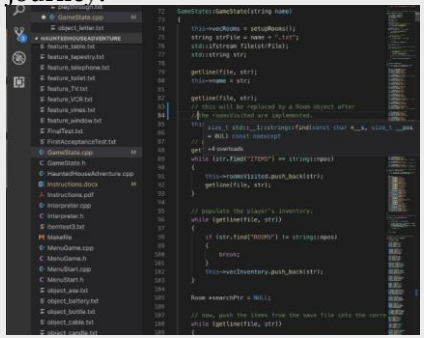
Project Highlights

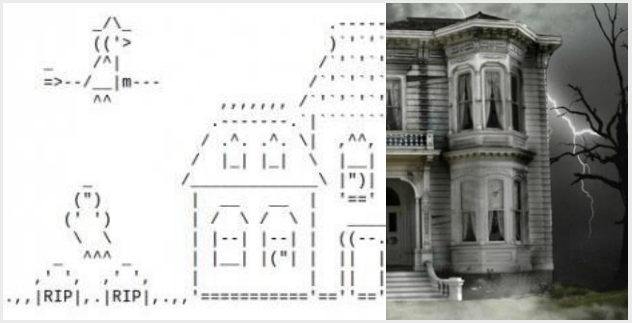
- Built-in text parser interprets user actions and intent, then validates and provides real-time feedback to the user based on the game environment.
- Game is built around haunted house lore and provides two distinct endings based on the choices that the user made during the game.
- Navigation engine responds *dynamically* to the game state and user commands instead of forcing the user into pre-built tracks.
- Codebase is designed with a modular, object-oriented architecture, allowing for easy maintainability and future expansion.
- Responses to user text input is programmatically created and presented to the user. The same command could present the user with completely different game feedback based on their current stage in the game, their inventory, or location.
- Rooms exist with their own independent states that persist throughout the entire gameplay session. Did you plug in the TV? Did you search the lockbox in the Conservatory? You'll see those changes in the game even after exploring the rest of the house and coming back.
- Game leverages ofstream, ifstream, and fstream libraries alongside an extensive library of custom game text. Want to adapt the Haunted House Adventure codebase for a game about a moon colony? Swap out the game text files with your own gameplay text without making any major code modifications. Just rename the rooms in the GameState class and have a brand new setting to explore.
- Algorithms for validating user input and routing the application to the correct interaction tracks were designed from scratch with flexibility and performance in mind.

Oregon State University

TBA II: Haunted House Adventure

An exciting journey through an abandoned house from the comfort of your own machine, written in C++. Explore the house, interact with its features, and load up your knapsack with items to help you on your journey.



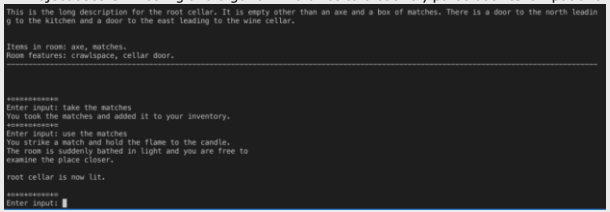


Gameplay Highlights

- In-game text parser expands user input options beyond just the simple "take the item." Seasoned tabletop and text game players will love the flexibility of text inputs provided to the user.
- 15 rooms for you to explore, each with their own unique features and opportunities for interactions.
- Help menu and in-game hints are available for new players, or players looking for ideas on how to discover new gameplay events.
- Flexible game saving and loading are always available. Haunted House Adventure supports as many save game files as your computer can hold!
- Haunted House Adventure goes beyond programmatically-generated text feedback. Experience the depth of object with their own characteristics and behaviors.

Technical Highlights

- Application written entirely with C++11 and the Standard Template Library.
- Project compiles instantly using G++ with the included Makefile.
- Interaction sequences leverage external files for text generation, save files, and room interactions. As a result, the code is completely modular, even allowing for complete modifications to be made to the game (such as expansions, new themes/plots/items) without ever touching the source code.
- Project uses C++ string and algorithm libraries to creatively parse user text input and route interaction sequences.



- Game engine completely encapsulates the underlying data structures, allowing developers to quickly create complex in-game events and actions without worrying about stability.
- Object-oriented design provides superior readability and maintainability, while also providing a self-documenting project structure.