

Haunted Dungeon

Final Report

OSU CS467 Winter 2017

Team Pictor

Andrew Bagwell
Jerry Hayes
Daniel Loughlin

Introduction

The Pictor Team has created an Interactive Fiction game called Haunted Dungeon that challenges the player to escape a dungeon by discovering rooms and their secrets while avoiding hidden dangers that lie within. Can you find the key, escape the dungeon, and survive Blinky and his gang of ghosts?

What we have now is a functioning game that can be won or lost by the player.

User's Perspective

A throwback to the classic text-based adventure of the 1970's and 1980's, the Haunted Dungeon evokes a sense of nostalgia for players of classics such as Zork and provides a window to the distant past of computer gaming for other, less familiar players. Similar to other games in the interactive fiction genre, players of Haunted Dungeon will move through different rooms exploring the contents, fending off the attacks of ghosts, and solving puzzles along the way. Users unfamiliar with basic interactive fictions games may be shocked by the lack of graphical sophistication, but we think this will be an educational experience for these users as they will appreciate how current game development has often "stood on the shoulders" of the work of prior eras. The primary objective of Haunted dungeon is to find the golden key hidden within the dark labyrinth that is filled with treasures and hazards and use the key to escape before ghosts that guard the dungeon ends the player's life. A secondary objective is to find hidden passageways that led to priceless treasures and tote them out of the dungeon.

System Architecture and Flow

The proposed system architecture for Haunted Dungeon consisted of three sections as illustrated in figure 1. One issue that was discovered during integration was the

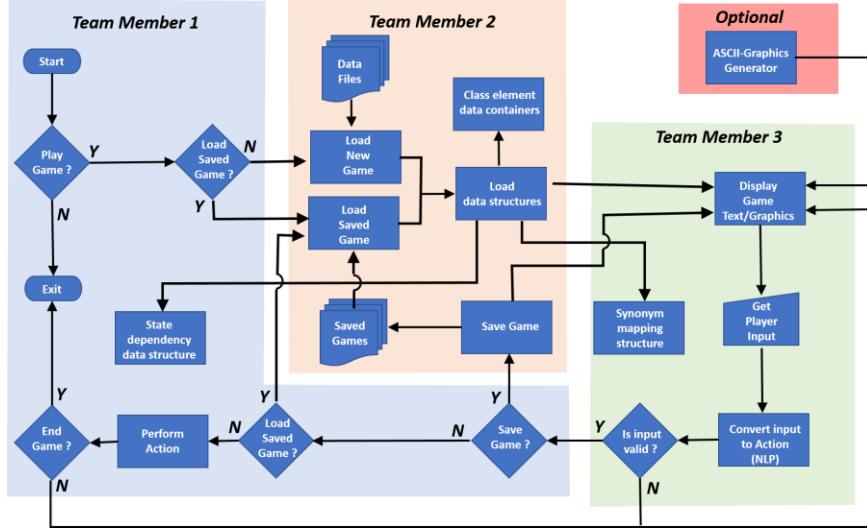


Figure 1: Initial software architecture proposed for Haunted Dungeon. Although changes were required in the final version, the basic functionality of the architecture was maintained. The most prevalent change was moving the display game text block to the top level.

awkward interaction of acquiring player input and displaying state dependent game responses back to the player. To eliminate this awkwardness, it was decided to i) extract the *Perform Action* block of team member 1 into its own module, ii) extract the *Convert Input to Action* and *Synonym Mapping* blocks of team member 3 into its own module, and iii) merge the remaining blocks of team members 1 and 3 into a top level file as shown in figure 2. The top level file was owned by both Andrew and Daniel which resulted in an XP pair programming like environment. The net result of the final architecture was a cleaner top level implementation and a clearer definition of responsibilities of the three game modules, i.e. the Natural Language Processing module, the Data Management module, and the Perform Action module. Note that the Perform Action module was expanded to include Ghost Combat which provides a random element to the game and controls outcomes of interactions between ghosts and the player.

After every player turn, a routine is run in the Perform Action module for each of the four ghosts in the game. The routine provides a random chance that a ghost becomes visible or moves to another room. However, if a player is in the same room, the ghost will attack. An attack has a chance to hit is lessened by a player's armor. The better armor a player has, the less likely to be hit for damage by a ghost. If the ghost succeeds in landing a blow, damage is deducted from the player's health. In response, a player can use a turn to attack a ghost. The requirements are that a player must be visible and have a weapon equipped. The weapons in the game all have a max damage attribute that is used as the upper bound when determining damage to a ghost.

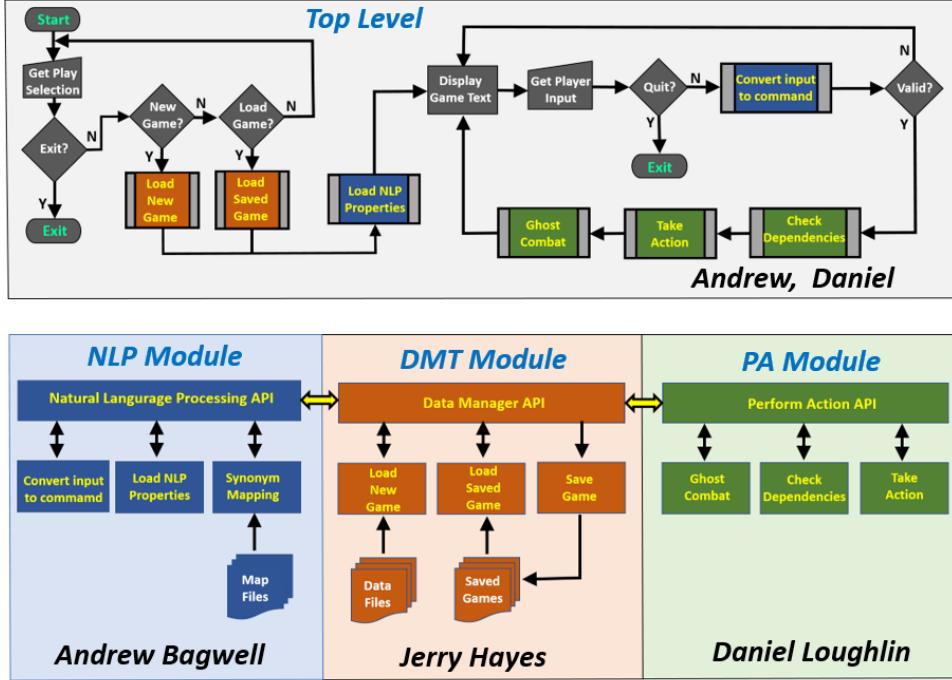


Figure 2: Final software architecture of Haunted Dungeon. The three modules, i.e. NLP, DMT, and PA, define class structures that are instantiated at the top level that provides services (methods) via their respective APIs. Note that the modules also access each other's services.

The Natural Language Processing (NLP) module uses a variety of algorithmic steps in converting the player's input to a supported command. There were four main objectives of the NLP module, i) Remove all extraneous words that does not appear in any commands or synonyms, ii) map synonyms to a supported game verb, iii) provide some tolerance to spelling errors, and iv) provide some tolerance to word order. To meet these four objectives, Andrew researched and experimented with multiple approaches, ultimately settling on approaches based on tokenization. Two methods were implemented using tokenization: `buildTuple()` and `matchTuple()`. Each method has different strengths and weakness when scored against the four objectives.

Both methods have high tolerance for spelling errors as they both call the same methods - `parseCommands()` and `cleanCommands()` - to parse data, remove extraneous symbols, and handle user spelling errors. The `parseCommands()` method parses the entire command string into a list of string tokens and returns that list. The `cleanCommands()` method uses the Levenshtein Distance technique to perform a fuzzy match between tokens and supported game words. This is done by computing a similarity ratio between a token and every supported word. If the highest similarity ratio for a token/gameword match exceeds a certain threshold (.7), that token is replaced by the gameword in that token list.

After that the two methods diverge. The buildTuple() method works as follows: (1) it calls the replaceSyns() method to replace synonyms of gamewords with the actual gameword, (2) two containers are created, words and trash, and each token is examined, (3) until a supported verb is found, every token is placed into the trash container, (4) if a verb is found, the verb is placed into the word container and the verb flag boolean is set , (5) if a preposition is found and the verb flag is set, the preposition is placed into the words container, (6) otherwise, if the verb flag is set, it add the token to words and if the verb flag is not set, it adds the token to trash, (7) if the verb flag is set, the tokens in the words container are combined to form a command tuple, (8) if not, the tokens in the trash container are combined to form a tuple, (8) the tuple is then returned to the caller.

The matchTuple() method works differently: (1) gets a list of every supported command tuple, (2) iterate through each tuple, (3) split the tuple into pieces, (4) compare each tuple piece to each command token, (5) if the tuple piece and the command token straight match, add 1 to the score for the tuple, (6) if not and if the tuple piece matches a synonym of the token, add 1 to the score for the tuple, (7) if there's not a straight match or match to a synonym, compute the similarity ratio using Levenshtein Distance technique, (8), if the similarity ratio exceeds the threshold set - separate threshold from that of cleanCommands - then add the similarity ratio to the score for the tuple, (9) return the tuple with the highest score to the caller.

The matchTuple() method provides more tolerance for word order than buildTuple(). Similarly, matchTuple() is more accommodating to many imprecise commands as it will attempt to match user input to the most similar supported command-tuple. However, this capability has a downside as seemingly logical, but non-supported user input can be matched to an irrelevant command tuple, potentially causing confusion to the user and adversely affecting the user experience. So, ultimately, we chose to use buildTuple() in production as it meets the four objectives well enough, interfaces better with the other modules, and provides a better gameplay experience for the user.

The save game block of figure 1 was eliminated in figure 2 because the commands “loadgame” and “savegame” are treated like any other of the 250+ game commands. A player’s input is first sent the NLP module which makes a best effort attempt to match it to one of the many supported commands. Once matched, the valid command is sent to the PA module which sequentially checks the dependencies associated with the command. The first dependency that is missing will cause the program to abort any action from being taken. If all dependencies are met the PA module will perform all actions associated with the command. Since the “loadgame” and “savegame” commands have no dependencies the PA module will always call their respective actions methods in the Data Management module. The dependencies and actions associated with every command can be found in the [dependency.json](#) file.

Team Member Accomplishments

Andrew Bagwell

Natural Language Processing Module: This module is responsible for processing user input into a supported “command tuple”. The module includes two approaches for doing so: buildTuple() and matchTuple(). Both approaches use a string matching technique that incorporates the Levenshtein Distance metric to account for spelling mistakes in the user input, have word order tolerance, and appropriately handle unsupported input commands.

Jerry Hayes

Data Management Module: The module is responsible for loading and saving game data, creating primitive class instances for all entity types used by the game, and providing an API that is used for querying and manipulating the state of the game.

Primary.json: Contains attribute information used to create instances of the player, inventory, room, exit, object, and ghost class. Short and long descriptions of rooms, exits, objects, and ghost are also defined in the file.

Dependency.json: Describes all dependencies and actions associated with game commands. Textual responses to each dependency and action are also defined.

Data Attribute Unit Testing: Used to check attribute values defined in primitives.json for every call class instance.

Data Method Unit Testing: Used to check API methods for accessing and changing attributes associated with the state of the game.

Data Dependency Integration Testing: Used to check the accuracy of the dependency.json file

Daniel Loughlin

Perform Action Module: Module for performing actions based on command tuples created by the Natural Language Processing Module. This also implements ghost actions and combat.

Main Game Class: The main game module responsible for presenting the game to the user. Allows for loading the game and controls the gameplay loop.

Software and Development Tools

The game was written exclusively in Python. We support Python 2.7 running on the OSU engineering server flip1.engr.oregonstate.edu. We do not utilize third party modules outside of those as part of the standard set with Python 2.7 (eg. os, re, sys).

Starting the Game

1. Download the game zip file from Canvas and copy it to a flip server.
2. Unzip all files contained in “PictorFinalProject.zip”.
command prompt> ***unzip PictorFinalProject.zip***
3. Go to the top level of the game directory structure
command prompt> ***cd pictor***
4. Start the game
command prompt> ***python haunted_dungeon.py***
5. Select ‘Start a New Game’ option. If you have previously saved a game you can use the ‘Load a Saved Game’ option. Type 1
HAUNTED DUNGEON! !
1) Start A New Game
2) Load a Saved Game
3) Exit
Enter your choice> ***1***
6. Begin game navigation (see Game Walkthrough section for quick step by step walkthrough on completing a simple game). For self-exploration of the game, refer to the Appendix for a list of supported commands, a game map, and command examples

Game Walkthrough

Text Display Format



General commands

- **attack {ghost name}**: At any time a ghost may appear and begin attacking the player. If the player has sufficient health points, they may choose to stand and fight the ghost(s) by issuing the command “attack {ghost name}” where ghost name = {"blinky", "clyde", "inky", "pinky"}. Sometimes it is better to run than fight. To run, try to go to a room not occupied by a ghost. Eventually the ghost will move on to another location. Be careful not to let your health points drop to zero. You can restore your health points by eating, drinking, sitting on or laying on objects. You can limit the damage received in a ghost attack by wearing armor or a helmet. You can even avoid ghost attacks altogether by wearing the invisibility cloak.
- **ghosts**: Used to get a sneak preview of where the ghosts are.
- **help**: Used to list all supported verbs
- **savegame**: Used to save the current state of the game to a file.
- **loadgame**: Used to load a previously saved game state.
- **inventory**: Lists items in the player's inventory

Example step by step game walkthrough

The table below is a step by step walkthrough using a subset of commands and rooms that are supported in Haunted Dungeon. For self-exploration, refer to the Appendix for supported commands, a game map, special objects, and examples of how to use the various commands.

Step	Command	Description
1	<i>look at machete</i> (Cave Entrance)	The “look at” command prints the long description of an object. This command can be used to find useful clues for game play.
2	<i>take machete</i> (Cave Entrance)	The “take” command is used to place objects that are acquirable into the player’s inventory. Added machete to your inventory
3	<i>equip machete</i> (Cave Entrance)	The “equip” command is used to activate an offensive or defensive object. You are now armed with the machete
4	<i>grab shale rocks</i> (Cave Entrance)	Common synonyms are mapped to supported game verbs, e.g. “grab” is mapped to “take”. Added shale rocks to your inventory
5	<i>south</i> (Cave Entrance)	Simple direction {east, west, south, north } command. The exit is locked. The goal of the game is to find the key to escape this door. You will need to continue to venture into the dungeon to find the key.
6	<i>west</i> (Cave Entrance)	The game detects when an exit does not exist. There is no exit going west
7	<i>unlock south</i> (Cave Entrance)	Some directions need a key to be unlocked. You don't have the right key
8	<i>go north</i> (Cave Entrance)	When an exit exists and is unlocked, a direction command will cause the player to move to another room. You entered the Passageway
9	<i>rolling shutter door</i> (Passage Way)	Some rooms are dark. You can exit the room using the description of an exit. You entered the Common Area
10	<i>look at bar stool</i> (Common Area)	The “look at” command gives a hint to sit on the bar stool.
11	<i>sit on bar stool</i> (Common Area)	Sitting on the bar stool indicates you have increased your health.
12	<i>look at portrait</i> (Common Area)	The “look at” command gives a hint that something might be behind the portrait.
13	<i>look behind portrait</i> (Common Area)	You see a note that has been hidden behind the portrait.
14	<i>take note</i> (Common Area)	Added note to your inventory
15	<i>go smoky glazed doors</i> (Common Area)	You entered the kitchen
16	<i>look on kitchen table</i> (Kitchen)	You now see a green apple, a kitchen knife, and some matches.
17	<i>take green apple</i> (Kitchen)	Added green apple to your inventory

18	eat green apple (Kitchen)	Some objects are edible. When eaten, the object is consumed (removed from the game) and your health is increased
19	take matches (Kitchen)	Added matches to your inventory
20	go south (Kitchen)	You entered the Banquet Hall
21	take earthenware (Banquet Hall)	Added earthenware to your inventory
22	look on dining table (Banquet Hall)	You now see a bottle of water and fine china
23	take fine china (Banquet Hall)	Added fine china to your inventory
24	take bottle of water (Banquet Hall)	Added bottle of water to your inventory
25	drink bottle of water (Banquet Hall)	Drinkable items require a bottle opener. You do not have a bottle opener to open the bottle of water.
26	east (Banquet Hall)	You entered the Hallway
27	south (Hallway)	You entered the Armory
28	look on trestle table (Armory)	You now see armor, helmet, sabre, and a warhammer
29	take sabre (Armory)	Added sabre to your inventory
30	take armor (room: Armory)	You are carrying too much, cannot take armor You need to make room in your inventory
31	drop shale rocks (Armory)	Shale rocks removed from inventory and placed in the room
32	drop fine china (Armory)	Fine china removed from inventory and placed in the room
33	drop earthenware (Armory)	Earthenware removed from inventory and placed in the room
34	take armor (Armory)	Added armor to your inventory
35	equip armor (Armory)	You are now wearing the armor. You now have some protection against ghost attacks
36	take helmet (Armory)	Added helmet to your inventory
37	equip helmet (Armory)	You are now wearing the helmet. You have some additional protection against ghost attacks
38	equip sabre (Armory)	You already have a weapon equipped. You will need to unequip the machete first.
39	unequip machete (Armory)	Machete has been unequipped
40	equip sabre (Armory)	Your are now armed with the sabre
41	south (Armory)	You entered the Sleeping Quarters
42	look at bunk bed (Sleeping Quarters)	The "look at" command gives several hints, 1) that the bed is comfortable enough to "lay on" and 2)

		there appears to be something “under” the bed.
43	lay on bunk bed (Sleeping Quarters)	You have improved your health
44	look under bunk bed (Sleeping Quarters)	You now see the red apple that was hidden under the bunk bed
45	look on round table (Sleeping Quarters)	You now see a brass lantern
46	take brass lantern (Sleeping Quarters)	Added brass lantern to your inventory
47	go old wooden door (Sleeping Quarters)	You entered the bedroom
48	light brass lantern (Bedroom)	The room is dark. To see objects in the room the player lights the brass lantern
49	look at safe (Bedroom)	It is a combination safe that requires the player to know the combination.
50	look inside safe (Bedroom)	Safe is locked
51	read note (Bedroom)	You now know the combination of the safe
52	unlock safe (Bedroom)	Safe has been unlocked
53	look inside safe (Bedroom)	You now see the gem
54	look at wooden shelves (Bedroom)	A hint is giving that something may be behind the shelves.
55	look behind wooden shelves (Bedroom)	You now see a wooden lever
56	look at wooden lever (Bedroom)	A hint that the lever can be pulled
57	pull wooden lever (Bedroom)	The wooden lever opens a secret panel that transports the player to the closet.
58	look inside armoire (Closet)	You now see a cloak
59	take cloak (Closet)	Added cloak to your inventory
60	equip cloak (Closet)	You are now invisible
61	look at iron lever (Closet)	Gives a hint that the iron lever has been pushed many times
62	push iron lever (Closet)	The iron lever opens a secret panel that transports the player to the bedroom
63	look inside cedar trunk (Bedroom)	You can now see the golden key.
64	take golden key (Bedroom)	Added golden key to your inventory. This is the key needed to escape the dungeon!
65	east (Bedroom)	You entered the Sleeping Quarters
66	north (Sleeping Quarters)	You entered the Armory. Notice that the objects you dropped earlier are still in the room.
67	rusting iron door (Armory)	You entered the Jail
68	look on desk	You see a bottle of gatorade and a bottleopener

	(Jail)	
69	take bottleopener (Jail)	Added bottleopener to your inventory
70	drink bottle of water (Jail)	Bottle of water is consumed and your health has increased.
71	west (Jail)	You entered the Armory
72	go north (Armory)	You entered the Hallway
73	monstrous archway (Hallway)	You entered the Banquet Hall
74	go splintered double doors (Banquet Hall)	You entered the Kitchen
75	go west (Kitchen)	You entered the Common Area
76	south (Common Area)	You entered the Passageway.
77	south (Passageway)	You entered the Cave Entrance
78	unlock corrugated steel door (Cave Entrance)	Corrugated steel door has been unlocked
79	go south (Cave Entrance)	You entered the Outside. Congratulations, you have escaped the Haunted Dungeon. You can continue to explore back in the dungeon, if you like.

Game Stats

- 1 player
- 1 inventory bag
- 17 rooms
- 20 verb-preposition combos
- 74 objects
- 32 acquirable objects
- 270 game commands
- 29 exits
- 3 secret passageways
- 4 ghosts

Summary

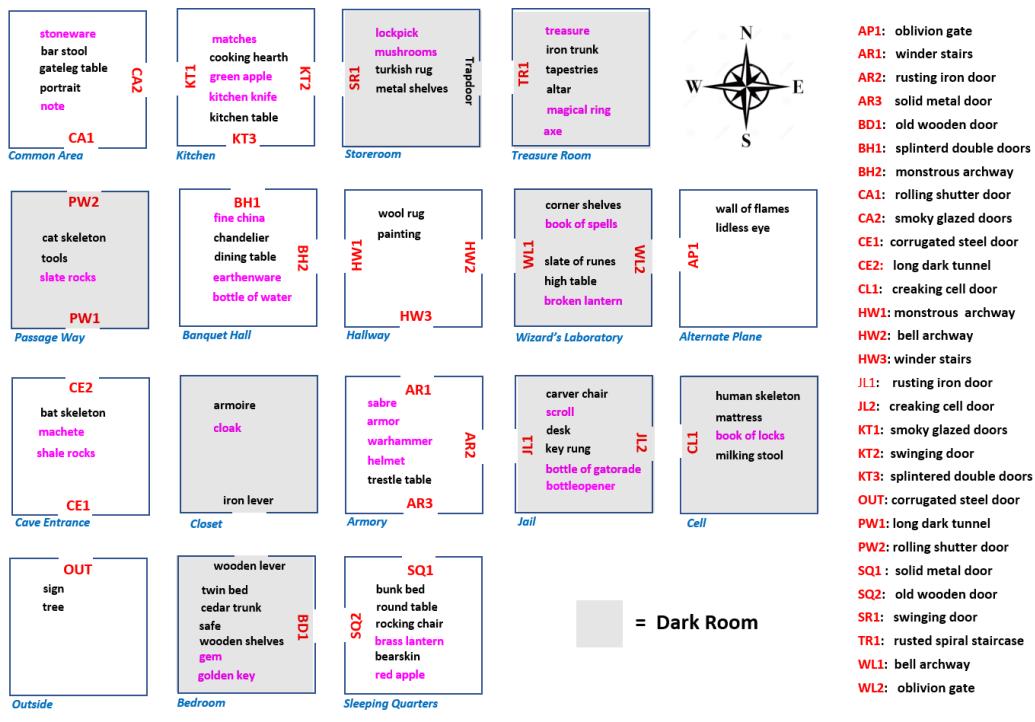
It is the opinion of team Pictor that this project was a success not only because we were able to meet all of the client's requirements, but also because of the great effort delivered by each team member. This can be seen when viewing the over [150 code commits](#) on GitHub that span the 8 week development cycle. Each week every team member delivered well above the 10 hour requirement for the course. Communication between team members occurred almost daily which allowed ideas to be shared and advanced quickly. Although command line adventure games like Haunted Dungeon did not expose the team to any bleeding edge technologies, it did give each team member more practice with Python and was ultimately a great team experience.

Appendix

Supported Game Commands

ATTACK	equip sabre	look at gateleg table	LOOK BEHIND	sit on rocking chair
attack blinky	equip warhammer	look at gem	look behind portrait	sit on twin bed
attack clyde	GO	look at golden key	look behind wooden shelves	TAKE
attack inky	go west	look at green apple	LOOK INSIDE	take armor
attack pinky	go north	look at helmet	look inside armoire	take axe
DRINK	go east	look at high table	look inside cedar trunk	take book of locks
drink bottle of gatorade	so south	look at human skeleton	look inside desk	take book of spells
drink bottle of water	go {exit description}	look at iron lever	look inside iron trunk	take bottle of gatorade
DROP	west	look at iron trunk	look inside safe	take bottle of water
drop armor	north	look at key rung	LOOK ON	take bottleopener
drop axe	east	look at kitchen knife	look on altar	take brass lantern
drop book of locks	south	look at kitchen table	look on corner shelves	take broken lantern
drop book of spells	{exit description}	look at lidless eye	look on desk	take cloak
drop bottle of gatorade	LAY ON	look at lockpick	look on dining table	take earthenware
drop bottle of water	lay on altar	look at machete	look on gateleg table	take fine china
drop bottleopener	lay on bearskin	look at magical ring	look on high table	take gem
drop brass lantern	lay on bunk bed	look at matches	look on kitchen table	take golden key
drop broken lantern	lay on mattress	look at mattress	look on mattress	take green apple
drop cloak	lay on turkish rug	look at metal shelves	look on metal shelves	take helmet
drop earthenware	lay on twin bed	look at milking stool	look on round table	take kitchen knife
drop fine china	lay on wool rug	look at mushrooms	look on trestle table	take lockpick
drop gem	LIGHT	look at note	look on wooden shelves	take machete
drop golden key	light brass lantern	look at painting	LOOK UNDER	take magical ring
drop green apple	LOOK ABOVE	look at portrait	look under bearskin	take matches
drop helmet	look above altar	look at red apple	look under bunk bed	take mushrooms
drop kitchen knife	LOOK AT	look at rocking chair	look under mattress	take note
drop lockpick	look at altar	look at round chair	look under turkish rug	take red apple
drop machete	look at armoire	look at sabre	look under wool rug	take sabre
drop magical ring	look at armor	look at safe	MISCELLANEOUS	take scroll
drop matches	look at axe	look at scroll	ghosts	take shale rocks
drop mushrooms	look at bar stool	look at shale rocks	help	take slate rocks
drop note	look at bat skeleton	look at sign	inventory	take stoneware
drop red apple	look at bearskin	look at slate of runes	loadgame	take treasure
drop sabre	look at book of locks	look at slate rocks	look	take warhammer
drop scroll	look at book of spells	look at stoneware	savegame	UNEQUIP
drop shale rocks	look at bottle of gatorade	look at tapestries	PULL	unequip armor
drop slate rocks	look at bottle of water	look at tools	pull wooden lever	unequip axe
drop stoneware	look at bottleopener	look at trapdoor	pull trapdoor	unequip cloak
drop treasure	look at brass lantern	look at treasure	PUSH	unequip helmet
drop warhammer	look at broken lantern	look at tree	push iron lever	unequip kitchen knife
EAT	look at bunk bed	look at trestle table	READ	unequip machete
eat green apple	look at carver chair	look at turish rug	read book of locks	unequip magical ring
eat mushrooms	look at cat skeleton	look at twin bed	read book of spells	unequip sabre
eat red apple	look at cedar trunk	look at wall of flames	read note	unequip warhammer
EQUIP	look at chandelier	look at warhammer	read scroll	UNLOCK
equip armor	look at cloak	look at wooden lever	read sign	unlock west
equip axe	look at cooking hearth	look at wooden shelves	read slate of runes	unlock east
equip cloak	look at corner shelves	look at wool rug	SIT ON	unlock north
equip helmet	look at desk	look at blinky	sit on bar stool	unlock south
equip kitchen knife	look at dining table	look at clyde	sit on bunk bed	unlock iron trunk
equip machete	look at earthenware	look at inky	sit on carver chair	unlock safe
equip magical ring	look at fine china	look at pinky	sit on milking stool	unlock corrugated steel door

Game Map



Magical Objects

- cloak:** A magical artifact that renders the wearer invisible when equipped.
- magical ring:** A magical artifact that when equipped increases the damage potential of an equipped weapon.

Game Villains



Considered the leader of the ghost gang, Blinky is the toughest and most lethal ghost in the game. He randomly traverses all the rooms in the game and switches between visible and non-visible modes depending upon his mood. He does not actively pursue the player, however when visible and in the same room as the player he is a force to be reckoned with.



Although not as lethal as Blinky, Clyde is still a major threat to the player. He is both tough to kill and deadly in his attacks against the player. Clyde restricts himself to a subset of the rooms in the game and has a higher potential of being visible than Blinky. Since ghosts can only attack when uncloaked, the player has a higher probability of engaging with Clyde than with Blinky.



Pinky is the most aggressive ghost in the game, fortunately for the player she is also the weakest and least deadly. She often positions herself in either the room the player is in or an adjoining room hoping to ambush the player when he comes in. She can be found in any of the rooms



Inky is the most fickle of all the ghost. Sometimes she feels aggressive and other times not so much. She tends to restrict herself to a subset of the rooms and will either try to flee from the player or fight, depending upon her mood. She is more powerful than Pinky, however not as tough or lethal as either Clyde or Blinky.

Command Examples

Attack {ghost}



Attack Inky: The player can choose to stand and fight if they believe they have enough health points to survive the battle. To kill Inky, the player must reduce the ghost's health points to zero by issuing the command "attack Inky". The outcome of the attack updates the player's and ghost health points.

Previous State	Current State
Your move: north (Previous command)	Your move: attack inky (Current command)

You entered the Armory (Previous result)	(Player lands a blow on Inky)

HEALTH: 100 PROTECTION: 0 EQUIPPED WEAPON: machete DAMAGE POTENTIAL: 15	FIGHT: You landed a mighty blow dealing 7 damage! (Inky lands a blow on the player) FIGHT: Inky attacked you and reduced your health to 99

INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 25 INVENTORY ITEMS: [machete] Need an equipped weapon EQUIPPED ITEMS: [machete] that has damage potential	HEALTH: 99 (Player's health reduced) PROTECTION: 0 EQUIPPED WEAPON: machete DAMAGE POTENTIAL: 15

ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall	INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 25 INVENTORY ITEMS: [machete] EQUIPPED ITEMS: [machete]

You see these items ... * trestle table	ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall

OH NO!!! You see a ghost! It's Inky	You see these items ... * trestle table

Your move: attack inky (Current command)	OH NO!!! You see a ghost! It's Inky

Your move:	Your move:

Drink {object}



Drink Bottle of Gatorade: To drink the bottle of gatorade, the player must first have it and a bottleopener in their inventory. The player then issues the command “drink bottle of gatorade” which consumes the gatorade and increases the player’s health.

Previous State	Current State
<p>Your move: north (Previous command)</p> <hr/> <p>You entered the Hallway (Previous result)</p> <hr/> <p>HEALTH: 100 PROTECTION: 0</p> <p>ITEMS: [bottle of gatorade, bottleopener]</p> <p>(Item to be drank)</p> <p>ROOM: Hallway EXIT: 'Monstrous archway' on the west wall EXIT: 'Bell archway' on the east wall EXIT: 'Wnder stairs' on the south end of the room</p> <p>You see these items... * wool rug * painting</p> <p>Your move: drink bottle of gatorade (Current command)</p>	<p>Your move: drink bottle of gatorade (Current command)</p> <hr/> <p>Bottle of gatorade is consumed</p> <p>Your health has increased!</p> <hr/> <p>HEALTH: 135 (Player's health has increased) PROTECTION: 0</p> <p>ITEMS: [bottleopener]</p> <p>(Bottle of gatorade is removed from inventory)</p> <p>ROOM: Hallway EXIT: 'Monstrous archway' on the west wall EXIT: 'Bell archway' on the east wall EXIT: 'Wnder stairs' on the south end of the room</p> <p>You see these items... * wool rug * painting</p> <p>Your move:</p>

Drop {object}



Drop Shale Rocks: The player can choose to remove objects from inventory in order to make room for other objects. To remove the shale rocks, they must first be in the player’s inventory and then the player issues the command “drop shale rocks”. They are then removed from inventory and placed in the room.

Previous State	Current State
<p>Your move: south (Previous command)</p> <hr/> <p>You entered the Cave Entrance (Previous result)</p> <hr/> <p>HEALTH: 100 PROTECTION: 0</p> <p>ITEMS: [machete, shale rocks]</p> <p>ROOM: Cave Entrance EXIT: 'Corrugated steel door' on the south wall EXIT: 'Long dark tunnel' at the north end of the cave</p> <p>You see these items... * bat skeleton</p> <p>Your move: drop shale rocks (Current command)</p>	<p>Your move: drop shale rocks (Current command)</p> <hr/> <p>Shale rocks removed from your inventory</p> <p>Shale rocks placed in room</p> <hr/> <p>HEALTH: 100 PROTECTION: 0</p> <p>ITEMS: [machete]</p> <p>ROOM: Cave Entrance EXIT: 'Corrugated steel door' on the south wall EXIT: 'Long dark tunnel' at the north end of the cave</p> <p>You see these items... * bat skeleton * shale rocks (shale rocks is now visible in room)</p> <p>Your move:</p>

Eat {object}



Eat Green Apple: To eat the green apple, the player must first have it in their inventory. The player then issues the command "eat green apple" which consumes the green apple (removes it from the game) and increases the player's health

Previous State	Current State
<pre>Your move: take green apple (Previous command) ----- Added green apple to your inventory (Previous result) ----- HEALTH: 100 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 30 INVENTORY ITEMS: [machete, green apple] (Item to be eaten) ROOM: Kitchen EXIT: 'Smoky glazed doors' on the west wall EXIT: 'Swinging door' on the east wall EXIT: 'Splintered double doors' on the south wall You see these items... * matches * cooking hearth * kitchen knife * kitchen table Your move: eat green apple (Current command)</pre>	<pre>Your move: eat green apple (Current command) ----- Green apple is consumed Your health has increased! ----- HEALTH: 115 (Player's health has increased) PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 25 (Inventory weight has decreased) INVENTORY ITEMS: [machete] (Green apple is removed from inventory) ROOM: Kitchen EXIT: 'Smoky glazed doors' on the west wall EXIT: 'Swinging door' on the east wall EXIT: 'Splintered double doors' on the south wall You see these items... * matches * cooking hearth * kitchen knife * kitchen table Your move:</pre>

Equip {weapon}



Equip Machete: The player can equip the machete if the machete is in the player's inventory and no other weapon is currently equipped. The player equips the machete by issuing the command "equip machete". Once equipped, the machete can be used to attack ghosts.

Previous State	Current State
<pre>Your move: unequip sabre (Previous command) ----- Sabre has been unequipped (Previous result) ----- HEALTH: 100 PROTECTION: 20 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 135 INVENTORY ITEMS: [machete, armor, sabre] EQUIPPED ITEMS: [armor] ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall You see these items... * bunk bed * round table * rocking chair * bearskin Your move: equip machete (Current command)</pre>	<pre>Your move: equip machete (Current command) ----- You are now armed with the machete ----- HEALTH: 100 PROTECTION: 20 EQUIPPED WEAPON: machete DAMAGE POTENTIAL: 15 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 135 INVENTORY ITEMS: [machete, armor, sabre] EQUIPPED ITEMS: [machete, armor] (Machete is now equipped) ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall You see these items... * bunk bed * round table * rocking chair * bearskin Your move:</pre>

Equip {clothing}



Equip Helmet: The player can equip the helmet if the helmet is in the player's inventory. All equipped clothing items adds to the protection level the player has against ghost attacks. To equip the helmet, the player issues the command "equip helmet".

Previous State	Current State
<p>Your move: north (Previous command)</p> <hr/> <p>You entered the Armory (Previous result)</p> <hr/> <p>HEALTH: 100 PROTECTION: 20</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 140 INVENTORTY ITEMS: [sabre, armor, helmet] EQUIPPED ITEMS: [armor]</p> <p>ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall</p> <p>You see these items... * warhammer * trestle table</p> <p>Your move: equip helmet (Current command)</p>	<p>Your move: equip helmet (Current command)</p> <hr/> <p>You are now wearing the helmet</p> <p>(Protection level is increased)</p> <hr/> <p>HEALTH: 100 PROTECTION: 25</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 140 INVENTORTY ITEMS: [sabre, armor, helmet] EQUIPPED ITEMS: [armor, helmet]</p> <p>ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall</p> <p>You see these items... * warhammer * trestle table</p> <p>Your move:</p>

Lay on {object}



Lay on bunk bed: When the player is low on health points, he can restore his health by laying on certain objects. When in the sleeping quarters, the player can restore his health by issuing the command "lay on bunk bed" that results in an increase in his health. .

Previous State	Current State
<p>Your move: east (Previous command)</p> <hr/> <p>You entered the Sleeping Quarters (Previous result)</p> <hr/> <p>HEALTH: 100 PROTECTION: 0</p> <p>ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall</p> <p>You see these items... * bunk bed * round table * rocking chair * bearskin</p> <p>Your move: lay on bunk bed (Current command)</p>	<p>Your move: lay on bunk bed (Current command)</p> <hr/> <p>You have improved your health</p> <p>(Player's health increases)</p> <hr/> <p>HEALTH: 110</p> <p>ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall</p> <p>You see these items... * bunk bed * round table * rocking chair * bearskin</p> <p>Your move:</p>

Light {object}



Light Brass Lantern: The brass lantern can be used to light a dark room in order to see objects within the room. To light the brass lantern, both the lantern and the matches must first be in the player's inventory. The lantern can then be lit using the command "light brass lantern".

Previous State	Current State
<p>Your move: Old wooden door (Previous command)</p> <hr/> <p>You entered the Bedroom (Previous result)</p> <hr/> <p>HEALTH: 93 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern]</p> <p>ROOM: Bedroom EXIT: 'Old wooden door' on the east wall</p> <p>You do not see any items (Bedroom is dark)</p> <p>Your move: light brass lantern (Current command)</p>	<p>Your move: light brass lantern (Current command)</p> <hr/> <p>The brass lantern is lit (Current result)</p> <hr/> <p>HEALTH: 93 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern]</p> <p>ROOM: Bedroom EXIT: 'Old wooden door' on the east wall</p> <p>You see these items ...</p> <ul style="list-style-type: none">* twin bed* safe* wooden shelves* cedar trunk <p>Items are now visible because the room is now lit.</p> <p>Your move:</p>

Look above {object}



Look above altar: Some objects hide other objects above them. The "look at {object}" command will give hints if the object is hiding another object. The altar hides the axe above it and is not visible until the player issues the command "look above altar"

Previous State	Current State
<p>Your move: pull trapdoor (Previous command)</p> <hr/> <p>You are transported to the Treasure Room! (Previous result)</p> <hr/> <p>HEALTH: 106 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [brass lantern, matches]</p> <p>ROOM: Treasure Room EXIT: 'Rusted spiral staircase' on the west wall</p> <p>You see these items... * iron trunk * tapestries * altar * magical ring</p> <p>Your move: look above altar (Current command)</p>	<p>Your move: look above altar (Current command)</p> <hr/> <p>You see the following items(s): axe</p> <hr/> <p>HEALTH: 106 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [brass lantern, matches]</p> <p>ROOM: Treasure Room EXIT: 'Rusted spiral staircase' on the west wall</p> <p>You see these items... * iron trunk * tapestries * altar * magical ring * axe (Axe that was hiding above the altar is now visible)</p> <p>Your move:</p>

Look at {object}



Look at machete: To get more information about an object, the player can issue a “look at {object}” command which gives a long description of the object. For example, when the player issues “look at machete” the long description of the object is displayed.

Previous State	Current State
<pre>Your move: south (Previous command) ----- You entered the Cave Entrance (Previous result) ----- HEALTH: 100 PROTECTION: 0 ROOM: Cave Entrance EXIT: 'Corrugated steel door' on the south wall EXIT: 'Long dark tunnel' at the north end of the cave You see these items... * bat skeleton * machete * shale rocks Your move: look at machete (Current command)</pre>	<pre>Your move: look at machete (Current command) ----- A large heavy knife used for clearing underbrush that can also be used as a weapon when equipped (Object description) ----- HEALTH: 100 PROTECTION: 0 ROOM: Cave Entrance EXIT: 'Corrugated steel door' on the south wall EXIT: 'Long dark tunnel' at the north end of the cave You see these items... * bat skeleton * machete * shale rocks Your move</pre>

Look behind {object}



Look Behind Portrait: Multiple objects hide other objects behind them. The “look at {object}” command will give hints if the object is hiding another object. The portrait hides a note behind it. The note is not visible until the player issues the command “look behind portrait”

Previous State	Current State
<pre>Your move: west (Previous command) ----- You entered the Common Area (Previous result) ----- HEALTH: 100 PROTECTION: 0 ROOM: Common Area EXIT: 'Rolling shutter door' on the south wall EXIT: 'Smoky glazed doors' on the east wall You see these items... * bar stool * gateleg table * portrait Your move: look behind portrait (Current command)</pre>	<pre>Your move: look behind portrait (Current command) ----- You see the following items(s): note (Current result) ----- HEALTH: 100 PROTECTION: 0 ROOM: Common Area EXIT: 'Rolling shutter door' on the south wall EXIT: 'Smoky glazed doors' on the east wall You see these items... * bar stool * gateleg table * portrait * note (Note that was hidden behind portrait is now visible) Your move:</pre>

Look inside {object}



Look inside cedar trunk: Some objects hide other objects inside them. The “look at {object}” command will give hints if the object is hiding another object. The cedar trunk hides the golden key inside it and is not visible until the player issues the command “look inside cedar trunk”

Previous State	Current State
Your move: west (Previous command) ----- You entered the Bedroom (Previous result) ----- HEALTH: 75 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORTY ITEMS: [matches, brass lantern] ROOM: Bedroom EXIT: 'Old wooden door' on the east wall You see these items... * twin bed * safe * wooden shelves * cedar trunk Your move: look inside cedar trunk (Current command)	Your move: look inside cedar trunk (Current command) ----- You see the following items(s): golden key ----- HEALTH: 75 PROTECTION: 0 (Golden key found inside cedar trunk) INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORTY ITEMS: [matches, brass lantern] ROOM: Bedroom EXIT: 'Old wooden door' on the east wall You see these items ... * twin bed * safe * wooden shelves * cedar trunk * golden key (Golden key is now visible) Your move:

Look on {object}



Look on Round Table : Multiple objects hide other objects on them. The “look at {object}” command will give hints if the object is hiding another object. The round table hides the brass lantern on it. The brass lantern is not visible until the player issues the command “look on round table”

Previous State	Current State
Your move: go old wooden door (Previous command) ----- You entered the Sleeping Quarters (Previous result) ----- HEALTH: 100 PROTECTION: 0 ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall You see these items... * bunk bed * round table * rocking chair * bearskin Your move: look on round table (Current command)	Your move: look on round table (Current command) ----- You see the following items(s): brass lantern ----- HEALTH: 100 PROTECTION: 0 ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall You see these items... * bunk bed * round table * rocking chair * brass lantern — The brass lantern is now visible * bearskin Your move:

Look under {object}



Look under Turkish rug: Some objects hide other objects under them. The "look at {object}" command will give hints if the object is hiding another object. The Turkish rug hides the trapdoor under it and is not visible until the player issues the command "look under turkish rug"

Previous State	Current State
Your move: east (Previous command) ----- You entered the Storeroom (Previous result) ----- HEALTH: 75 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Storeroom EXIT: 'Swinging door' on the west wall You see these items... * turkish rug * metal shelves Your move: look under turkish rug (Current command)	Your move: look under turkish rug (Current command) ----- You see the following item(s): trapdoor ----- HEALTH: 75 PROTECTION: 0 (A trapdoor found under turkish rug) INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Storeroom EXIT: 'Swinging door' on the west wall You see these items... * turkish rug * metal shelves * trapdoor (Trapdoor is now visible) Your move:

Pull {object}



Pull Trapdoor: Some objects represent secret passageways to hidden rooms in the Haunted Dungeon. The trapdoor is one such object. When the player issues the command "pull trapdoor", the player is transported to the hidden treasure room.

Previous State	Current State
Your move: east (Previous command) ----- You entered the Storeroom (Previous result) ----- HEALTH: 75 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Storeroom EXIT: 'Swinging door' on the west wall You see these items... * turkish rug * metal shelves * trapdoor Your move: pull trapdoor (Current command)	Your move: pull trapdoor (Current command) ----- You are transported to the Treasure Room! ----- HEALTH: 75 PROTECTION: 0 (Trapdoor is secret access to treasure) INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Treasure Room (Player is now in the treasure room) EXIT: 'Rusted spiral staircase' on the west wall You see these items... * iron trunk * tapestries * altar Your move:

Push {object}



Push iron lever: Some objects represent secret passageways to rooms in the Haunted Dungeon. The iron lever is one such object. When the player issues the command “push iron lever”, the player is transported to the bedroom.

Previous State	Current State
<pre>Your move: look at iron lever (Previous command) ----- An iron bar that looks to have been pushed many times (Previous result) HEALTH: 82 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Closet You see these items... * armoire * iron lever Your move: push iron lever (Current command)</pre>	<pre>Your move: push iron lever (Current command) ----- You are transported to the Bedroom! (Your move) HEALTH: 82 PROTECTION: 0 Iron lever provides access to bedroom from closet INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORY ITEMS: [matches, brass lantern] ROOM: Bedroom (Player is now in the bedroom) EXIT: 'old wooden door' on the east wall You see these items... * twin bed * safe * wooden shelves * wooden lever * cedar trunk Your move:</pre>

Read {object}



Read Note: The note behind the portrait can be read which provides clues useful for game play. To read the note, the note must first be in the player's inventory and the player must be in a lighted room. The player then issues the command “read note” which provides the combination to the safe.

Previous State	Current State
<pre>Your move: take note (Previous command) ----- Added note to your inventory (Previous result) HEALTH: 100 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 1 INVENTORY ITEMS: [note] (Item has to be in inventory) ROOM: Common Area EXIT: 'Rolling shutter door' on the south wall EXIT: 'Smoky glazed doors' on the east wall You see these items... * bar stool * gateleg table * portrait Your move: read note (Current command)</pre>	<pre>Your move: read note (Current command) ----- You now know the combination to the safe! (Your move) HEALTH: 100 PROTECTION: 0 INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 1 INVENTORY ITEMS: [note] ROOM: Common Area EXIT: 'Rolling shutter door' on the south wall EXIT: 'Smoky glazed doors' on the east wall You see these items... * bar stool * gateleg table * portrait Your move:</pre>

Sit on {object}



Sit on rocking chair: When the player is low on health points, he can restore his health by sitting on certain objects. When in the sleeping quarters, the player can restore his health by issuing the command "sit on rocking chair" that results in an increase in his health. .

Previous State	Current State
<p>Your move: east (Previous command)</p> <p>-----</p> <p>You entered the Sleeping Quarters (Previous result)</p> <p>-----</p> <p>HEALTH: 82 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORTY ITEMS: [matches, brass lantern]</p> <p>ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall</p> <p>You see these items... * bunk bed * round table * rocking chair * bearskin</p> <p>Your move: sit on rocking chair (Current command)</p>	<p>Your move: sit on rocking chair (Current command)</p> <p>-----</p> <p>You catch your breath ... gain some health</p> <p>-----</p> <p>HEALTH: 87 (Sitting increased the players health) PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORTY ITEMS: [matches, brass lantern]</p> <p>ROOM: Sleeping Quarters EXIT: 'Solid metal door' on the north wall EXIT: 'Old wooden door' on the west wall</p> <p>You see these items ... * bunk bed * round table * rocking chair * bearskin</p> <p>OH NO!!! You see a ghost! It's Inky</p> <p>Your move:</p>

Take {object}



Take bottle of water: The player can chose to take objects from the room if there is space available in the player's inventory. In order to place the bottle of water into inventory, the player issues the command "take bottle of water" and the bottle of water is placed in inventory and removed from the room.

Previous State	Current State
<p>Your move: look on dining table (Previous command)</p> <p>-----</p> <p>You see the following items(s): bottle of water, fine china (Previous result)</p> <p>-----</p> <p>HEALTH: 83 PROTECTION: 0</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 16 INVENTORTY ITEMS: [matches, brass lantern]</p> <p>ROOM: Banquet Hall EXIT: 'Splintered double doors' on the north wall EXIT: 'Monstrous archway' on the east wall</p> <p>You see these items... * fine china * chandelier * dining table * earthenware * bottle of water</p> <p>Your move: take bottle of water (Current command)</p>	<p>Your move: take bottle of water (Current command)</p> <p>-----</p> <p>Added bottle of water to your inventory</p> <p>-----</p> <p>HEALTH: 83 PROTECTION: 0 (Inventory weight has increased)</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 21 INVENTORTY ITEMS: [matches, brass lantern, bottle of water]</p> <p>ROOM: Banquet Hall EXIT: 'Splintered double doors' on the north wall EXIT: 'Monstrous archway' on the east wall</p> <p>You see these items ... * fine china * chandelier * dining table * earthenware</p> <p>Your move:</p>

Unequip {object}



Unequip Sabre: The player sometimes needs to unequip a weapon in order to equip a more powerful weapon. If the player wants to equip the Warhammer while they have the sabre equip, they must first issue the command "unequip sabre".

Previous State	Current State
<p>Your move: equip warhammer (Previous command)</p> <p>You already have a weapon equipped (Previous result)</p> <p>HEALTH: 100 (Sabre must first be unequip before the PROTECTION: 20 / Warhammer can be equipped) EQUIPPED WEAPON: sabre DAMAGE POTENTIAL: 20</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 160 INVENTORY ITEMS: [armor, sabre, warhammer] EQUIPPED ITEMS: [armor, sabre]</p> <p>ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall</p> <p>You see these items... * helmet * trestle table</p> <p>Your move: unequip sabre (Current command)</p>	<p>Your move: unequip sabre (Current command)</p> <p>Sabre has been unequipped</p> <p>HEALTH: 100 (No weapons are equipped) PROTECTION: 20</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 160 INVENTORY ITEMS: [armor, sabre, warhammer] EQUIPPED ITEMS: [armor] (Sabre no longer equipped)</p> <p>ROOM: Armory EXIT: 'Winder stairs' going north EXIT: 'Rusting iron door' on the east wall EXIT: 'Solid metal door' on the south wall</p> <p>You see these items... * helmet * trestle table</p> <p>Your move:</p>

Unlock {object}



Unlock iron trunk: To look inside some objects they must first be unlocked. To unlock an object, the "key" to the object must first be in inventory. To know what key is needed the player issues "look at {object}" for clues. To unlock the iron chest the player must have a Lockpick then issue "unlock iron trunk".

Previous State	Current State
<p>Your move: look inside iron trunk (Previous command)</p> <p>Iron trunk is locked (To unlock the trunk , the player must have a Lockpick in their inventory.)</p> <p>HEALTH: 107 Hints are given when the player issues a PROTECTION: 20 "look at iron trunk")</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 96 INVENTORY ITEMS: [armor, brass lantern, lockpick] EQUIPPED ITEMS: [armor]</p> <p>ROOM: Treasure Room (Required "key" to unlock iron trunk) EXIT: 'Rusted spiral staircase' on the west wall</p> <p>You see these items... * iron trunk * tapestries * altar</p> <p>Your move: unlock iron trunk (Current command)</p>	<p>Your move: unlock iron trunk (Current command)</p> <p>Iron trunk has been unlocked</p> <p>HEALTH: 107 (The iron trunk is now unlocked. The player PROTECTION: 20 can now issue a "look inside iron trunk")</p> <p>INVENTORY CAPACITY: 200 INVENTORY WEIGHT: 96 INVENTORY ITEMS: [armor, brass lantern, lockpick] EQUIPPED ITEMS: [armor]</p> <p>ROOM: Treasure Room EXIT: 'Rusted spiral staircase' on the west wall</p> <p>You see these items ... * iron trunk * tapestries * altar</p> <p>Your move:</p>